University of Cape Town

Doctorate Thesis

Structured Incorporation of Model Uncertainty for Bayesian Adaptive Tracking and its Application to Maritime Surveillance

Author:
Charles Bradshaw

Supervisor:
Assoc. Prof. Fred Nicolls
Co-supervisor:
Prof. Gerhard de Jager

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Department of Electrical Engineering
University of Cape Town

December 2017

Declaration of Authorship

I, Charles Bradshaw, declare that this thesis titled, 'Structured Incorporation of Model Uncertainty for Bayesian Adaptive Tracking and its Application to Maritime Surveillance' and the work presented in it are my own. I confirm that:

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

Signed: _____

Date: _____

"I never trust anyone who is more excited about success than about doing the thing they want to be successful at."

Randall Munroe

Abstract

Structured Incorporation of Model Uncertainty for Bayesian Adaptive Tracking and its Application to Maritime Surveillance

by Charles Bradshaw

Adaptive visual object tracking (VOT) is one of the fundamental tasks in machine vision, with active research and far-reaching implications. Bayesian methods are commonly used in adaptive VOT. However, we propose that the current tendency is to restrict the inference to a subtask (e.g. classification), rather than phrasing the entire task, including the adaptive observation model, within the Bayesian inference. In this thesis we develop a framework for simultaneous modelling and estimation (SMAE), in which the common Bayesian recursive estimator (BRE) is extended to include estimation of the underlying hidden Markov model (HMM). The framework is developed not only for the task of adaptive VOT, but also for persistent tracking: the long-term task including automatic detection and tracking of multiple targets in a scene in a manner such that performance improves as a function of deployment time.

To prove that the framework is usable and leads to tractable implementations, it is applied to the challenging task of maritime surveillance. Oceans provide a non-trivial noisy background against which many adaptive trackers struggle. Our developed adaptive tracker creates a baseline in which the joint distribution across observation model and target state is maintained in an adapted particle filter. A persistent tracker is then built around the adaptive tracker to produce improved results using the information from previous observations. Both the adaptive tracker and the persistent tracker use the holistic Bayesian framework described by SMAE. We find that SMAE does lead to tractable solutions that include the strength of Bayesian methods for the observation model component in adaptive VOT. In addition to this, contributions are made to the current maritime surveillance literature, in the form of a better performing salience filter for maritime and littoral scenes, and a Bayesian means for combining different salience filters. This last contribution may seem trivial, however we were unable to find it in the maritime literature.

This work also includes the application of SMAE to more philosophical topics. Although the discussion may seem informal in light of the technical nature of the body of our work, it was an integral part of the development of the framework.

Acknowledgements

I would like to acknowledge the National Research Foundation and the University of Cape Town for funding this thesis.

My heartfelt gratitude goes to Prof Fred Nicolls, for all the advice and help on-topic and off-topic, for being able to make the thesis process seem tractable, while telling me I wasn't going to solve any problems, for his incredible insight, and for his seemingly endless patience with my comma-splices.

Thanks also to Prof Gerhard de Jager for his assistance while Prof Nicolls was on sabbatical in my first year, for letting me tackle the many varied topics that captured my interests.

To Becky, for supporting me through this process, for believing I would finish even when my velocity seemed zero, for life together, and for being willing to put our marriage to the test by proof-reading.

To all my friends and family for patiently accepting my refusal to admit I'd upgraded to a PhD, long after it was common knowledge.

Contents

List of Figures

List of Tables

Abbreviations

BRE	**B**ayesian **R**ecursive **E**stimator
FFT	**F**ast **F**ourier **T**ransform
GMM	**G**aussian **M**ixture **M**odel
HMM	**H**idden **M**arkov **M**odel
IR	**I**nfra-**R**ed
IRLS	**I**teratively **R**e-weighted **L**east **S**quares
LLR	**L**og **L**ikelihood **R**atio
MAP	**M**aximum **A** **P**osteriori
MHT	**M**ultiple **H**ypothesis **T**racker
MRF	**M**arkov **R**andom **F**ield
MTT	**M**ultiple **T**arget **T**racking
PDF	**P**robability **D**ensity **F**unction
SAR	**S**ynthetic **A**perture **R**adar
SNR	**S**ignal (to) **N**oise **R**atio
SMAE	**S**imultaneous **M**odelling **A**nd **E**stimation
VOT	**V**isual **O**bject **T**racking

Chapter 1

Introduction

Adaptive tracking is an active research field in which Bayesian frameworks are often used. However, the current literature appears to have a deficiency regarding a holistic Bayesian approach to the adaptive tracking problem, and it is this deficit that the current work addresses. In this chapter we present a summary of the document. The chapters that follow will provide detail, justification, and implementation for the various points made here.

This chapter starts off with section 1.1, establishing the concept that will underpin this work and presenting our hypothesis. Section 1.2 covers the context from which our research grows, leading into its purpose and scope. In section 1.3 we unpack the motivation behind, and the main features of, the framework that will be the key contribution of this work. Our choice of maritime surveillance as an application is covered in section 1.4, and the less concrete applications which we will discuss in the final chapter are established in section 1.5. We finish this chapter off with a summary of our novel contributions in section 1.6 and an overview of the rest of the document in section 1.7.

1.1 The Approach Spectrum

Most machine vision solutions can be seen as lying somewhere on spectrum that stretches from principled to practical (figure 1.1). The focus on the left is on accurately modelling the underlying problem, while approaches on the right are justified by "It works." Consider two solutions to tracking a red ball across a white background: the first solution is a Bayesian recursive estimator (BRE) implemented as a particle filter using template-matching for an observation model; the second is a graph-cut algorithm that separates the red voxels from the white voxels in the video sequence's space-time volume. Figure 1.2 illustrates the task and the solutions.

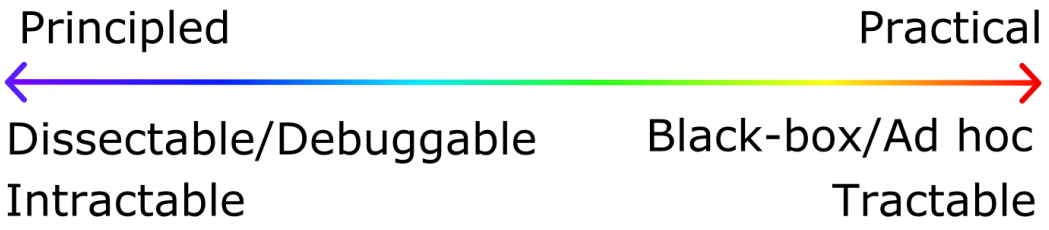
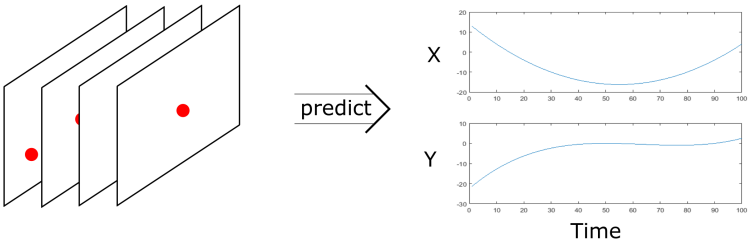


FIGURE 1.1: The approach spectrum. On the left we have approaches that attempt to model the problem as precisely as possible, and on the right we have solutions that try only to provide meaningful results. A principled approach tends to lead to approximate solutions to the exact problem, whereas a practical approach leads to exact solutions to an approximation of the problem.

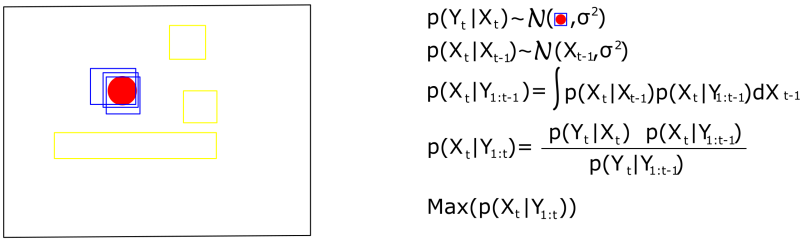
Even though the first solution contains approximations of the appearance model, the motion model, and the PDF, it is still modelling the underlying situation. The second solution will most likely produce excellent results, but there is a weak connection between the tracking problem and the voxel separation algorithm. Segmentations that are valid for voxels are not necessarily sane paths for an object through space-time. These two solutions represent the two sides of the spectrum.

While the first solution is an approximate solution to the exact problem, the second solution is an exact solution to an approximation of the problem. This is an important distinction. Algorithms always need re-factoring to improve or adapt to new problems. If a solution is an approximate solution to the exact problem, then its adjustments can be seen as moving through the space of all solutions. On the other hand, adjustments to an exact solution for an approximation of the problem move through the space of all problems.

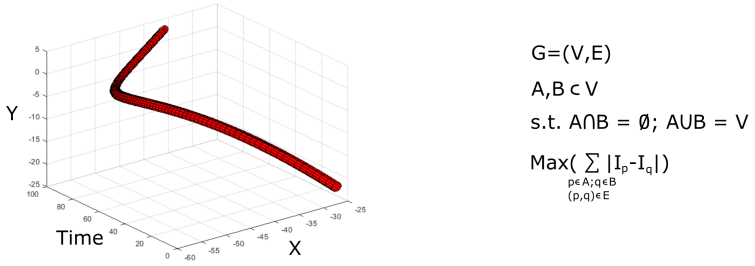
Consider changing the trackers to a more realistic problem with a cluttered background and a more detailed object. For the first solution, moving through the space of solutions, we can check the validity of the assumptions and approximations, and change the functional blocks accordingly. Moving through the space of problems is a far less tractable problem. The second solution relied on 'gimmicks' in the data. A designer would be forced to try define edge weights in the voxel space so that the graph cut still approximates the tracking task, and it is not guaranteed that this is possible. Additionally, debugging for the first solution involves comparing the output for the various stages to the outputs those stages would have if the solution were not an approximation, and investigating assumptions and approximations accordingly. Debugging for the second solution involves adjusting the parameters until the designer has a feel for what they each do. This is a far less certain process. Thus we can see that the first solution is



(A) A trivial tracking problem



(B) A principled solution



(C) A practical solution

FIGURE 1.2: A trivial tracking problem illustrating the difference between principled solutions and practical solutions. The task is to track a red ball across a white background as shown in (A). A principled approach to this task might lead to a Bayesian recursive estimator (as illustrated in (B)) using template-matching for an observation model and a simple Gaussian motion model. A practical approach to this task might lead to a graph-cut implementation on the space-time voxels (as illustrated in (C)), using an energy function to separate the red voxels from the white background. Both solutions are valid, yet they represent very different methodologies, and lead to very different design practices.

more generic, easier to adapt to new problems, and easier to debug[1]. The downside of principled solutions is a tendency to have larger processing requirements. Because they try to maintain a more accurate model of the problem, they tend to require more computing power, and can lead to intractable formulations.

This gives each side benefits and challenges. Few problems are simple enough that no heuristics would be required to model them in a tractable manner (hence the far left is ruled out), or that procedurally-generated algorithms would solve them (ruling out the far right). This means that designers face a choice. They can start on the left,

[1]To be clear, we are not speaking against dual problems. Solving a mathematically isomorphic problem is still mathematically solving the same problem, and hence lies on the left.

creating a principled framework that models the larger aspects of the problem, and then make approximations and use heuristics (i.e. lean to the right) as necessary towards a tractable solution. Alternatively, they can start on the right and lean left by connecting known components together in an ad hoc manner, and adjusting each component in isolation.

We propose that the current approaches to adaptive tracking fall in the latter category (a statement we will justify in section 1.3). Most approaches treat the learning of the observation model as a black-box inside a function approximator. Even if the tracker and classifier modules are both principled frameworks, current solutions characteristically lack a global framework that encompasses them both. This is understandable, as adaptive tracking is a very intricate problem to encompass entirely within an accurate model, and so solutions on the right have made advances. Our hypothesis is that it is possible to approach adaptive tracking in a manner that encompasses the entire task (including learning of the observation model) within a Bayesian framework and still leads to tractable solutions.

1.2 Project Context, Purpose and Scope

Bayesian methods have far-reaching implications and applications. The uninitiated hear of Bayesian methods and assume that the application of Bayes' rule is all that is being referred to, whereas in truth Bayesian thinking extends to almost every conceivable mental task. In his seminal work [2], Edwin Jaynes establishes probability as the extension of Aristotelian logic into a world with imperfect knowledge. Rather than reserving probabilities only for events that occur from repeatable trials, he shows that there is only one way a rational entity with a defined set of information can assign a value to an uncertain event in a manner that is consistent with what we consider common sense — a way that conforms to the rules of Bayesian probabilities. Thus he establishes probabilities not as a limit of repeated trials, but as an indicator of a rational entity's knowledge of a situation.

His presentation of the work is singular, and any summary we present will be deficient, yet we must proceed nonetheless. The central workhorse of Bayesian probabilities is the assignment of a representative value for a hypothesis X in light of observations Y as

$$p(X|Y) \propto p(Y|X)p(X). \tag{1.1}$$

Here the posterior $p(X|Y)$ is the only value that could be assigned to X in light of the observation Y by a rational entity consistent with common sense (as proved by

Jaynes [2]). The likelihood $p(Y|X)$ is the probability of event Y happening if X were true. The prior $p(X)$ is our acknowledgement that it is impossible to present a posterior without recognising our prior information, and the proportionality constant normalises the probabilities across all possible hypotheses.

This is all to say that Bayesian thinking epitomises the principled side of the approach spectrum. If a rational entity can only assign values to uncertain events in one way, then that way holds a monopoly on principle. We will assume that the reader is familiar with Bayesian reasoning. If our assumption is wrong, we refer the reader to Jaynes' work.

Bayesian methods are a staple of visual object tracking, and it is in the context of this field that we present our work. Many adaptive trackers use Bayesian frameworks such as the Bayesian recursive estimator (BRE), the Kalman filter, and the particle filter to handle the state estimation, yet we believe that there is a key approximation being made that they do not address. That is the ad hoc learning of the appearance model. We will justify this statement in section 1.3, but we reference it now because it is the inciting context for our project. Our purpose is to develop a principled Bayesian framework that encompasses both the tracking and the model learning in a tractable way.

There is a chance that in the sections to follow, it may appear as though we are advocating our method as the only 'correct' method. This is not the case. We work in a pragmatic field, and ultimately it is results that justify any work. There are valid reasons why authors have avoided a holistic Bayesian adaptive tracker. Our work here explores the hypothesis that a tractable, fully Bayesian, adaptive tracker can be envisioned and created.

The development of an untested framework is useless, so our scope will include both the development of the framework and its application on a real-world problem. In addition, during our work with the framework, we noted applications in modelling situations we face as humans. After we have derived and tested the framework, we will include these observations as an epilogue in chapter 7. They are not part of the scope of our work, but they are still useful and interesting applications of the framework.

1.3 The Call For Bayesian Adaptive Trackers

We pose the question, 'What makes a Bayesian tracker Bayesian?'

Does using a Kalman filter on the output make it Bayesian? Does framing one aspect of the problem as an Hidden Markov Model (HMM) make it Bayesian? Is it enough to use a Bayesian tool for an aspect of the task at hand? It can be. Each field defines what

words mean in its context. As authors, we use the word to imply more. We envisage a Bayesian adaptive tracker in which the entire task is encapsulated in a single monolithic inference: where the uncertainty in every variable is handled in a principled manner, and all observations are used.

In the paragraphs below, we point out attributes of particular trackers, and trends in the field that we believe do not epitomise this Bayesian approach. Our intention is not to cast judgement on the trackers we mention. Ours is a practical field, and success ultimately justifies any tracker. We draw attention to these Bayesian trackers, as they have proved the applicability of Bayesian methods to tracking, and in their ad hoc components we see the potential for a fully Bayesian approach.

Pèrez et al. [3] present a tracker that uses a particle filter to approximate the underlying BRE. For their observation model, they use a similarity measure between the color histogram of the candidate particle's bounding box and that of the initial frame. This limits the observations to include only the pixels within the bounding box. Later in the paper they include dividing this observation model by the similarity to the background, in contexts where it is possible to build a background model. This decision is presented as if motivated in an ad hoc way to improve results.

Zhang et al. [4] present a tracker that uses structured multi-task sparse learning (and is a generalisation of the L1 tracker by Mei and Ling [5]). The tracker uses a particle filter to approximate the BRE, and also considers the cropped bounding box as the observation. They reconstruct the resized patch as a linear combination of templates, and use the error of that reconstruction for the observation model. The templates are updated so as to keep those that are most relevant and useful in the template set.

Kwon and Lee [6] use a set of trackers that are combined in a way very similar to boosting, and whose particle filters interact. This is done using a BRE as justification, with each weak tracker providing an observation probability based on templates of previous tracking results and a motion model. The observation model represents only the cropped data. The fusing of the different sub-models is done as a weighted average, which while simple to code is a heuristic (the MIL tracker uses a more appropriate fusion model; we discuss it next). The tracker updates by including new trackers into the set of sub-trackers each time instant, with the new template cropped by the MAP estimate. This is done outside of the Bayesian framework, leaving us with no estimate on the uncertainty related to the new model.

The work closest to our own is the MIL tracker by Babenko et al. [7]. The main focus of their paper is the use of multiple instant learning (MIL) in labelling samples: a cunning way of cropping several templates from around the target and, without deciding which,

using the fact that at least one of them is the right cropping. The tracker uses a boosting formulation where each weak tracker is in the form of the log likelihood ratio (LLR):

$$h_k(x) = \log\left[\frac{p_t(y=1|f_k(x))}{p_t(y=0|f_k(x))}\right]. \tag{1.2}$$

Here the k-th classifier h_k uses the k-th Haar feature f_k, and both the positive and negative likelihoods are learned Gaussian distributions in the f_k dimension. While a Bayesian framework is not mentioned for the combined system, combining different classifiers by adding their LLRs is consistent with Bayesian reasoning for independent classifiers. We will show that using the LLR in this manner is equivalent to including the entire frame inside the observation, although this is not apparent in the text. Indeed, a Bayesian framework for the larger system is not presented. The learning of the foreground and background models, and the calculation for the boosting coefficient are justified by being functional. While a larger Bayesian framework may exist, it is not presented, and so one would not know what approximations and assumptions are being made in applying this framework.

The above are four examples, but the trend continues with other Bayesian trackers. There are two common traits in the literature: limiting the observations to the pixels inside the bounding box, and ad hoc observation model changes. It may seem unfair to point out these theoretical issues on working trackers, but we justify their importance below and more fully in chapter 3.

When a tracker either states explicitly or implies through its implementation that the observation Y_t will be the pixel values of the cropped candidate target (or features calculated from them), it butchers the assumption of the underlying HMM[2] in the BRE. In a HMM, the values of Y_t are dependent[3] on the unknowable yet definite value of the hidden variable X_t. These trackers make the values of Y_t dependent[4] on the BRE's choice of X_t. To put it another way: all rational observers should agree on the observation values. Labelling the bounding box's values as Y_t means that different considered states X_t will experience different observations Y_t. Enabling an inference engine to decide which values should be included in the observation is innately troubling. What makes this situation curious is that it is possible to construct a framework that is principled and justifies the current methods. If we pursue the principled approach (including the entire frame in Y_t), we will find it is equivalent to using a LLR for the cropped bounding box. Using a LLR seems intuitively a good idea — the tracker will be both pulling

[2]Out of concern for space in this chapter, we assume the reader is acquainted with the use of a hidden Markov model and Bayesian recursive estimation in visual object tracking. This is dealt with in a less accelerated manner in chapter 3.

[3]In the probability sense of the word.

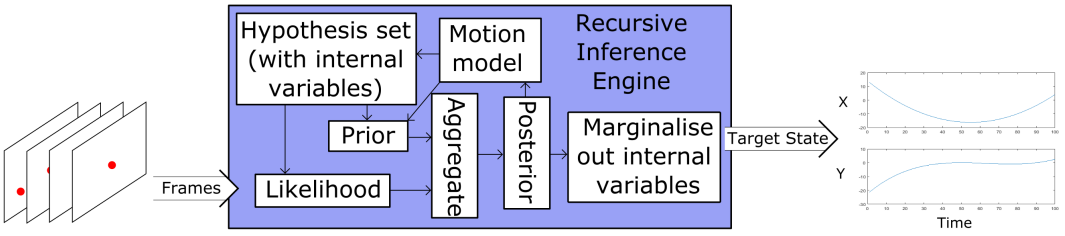
[4]That is, what is actually observed changes.

towards something favourable and pushing away from something negative, rather than only pulling. Many trackers use the LLR but still imply that Y_t is the cropped frame; we have not seen it shown as a consequence of the more appropriate definition of Y_t. Using the cropped bounding box as Y_t is consistent with an author using a Bayesian tool (that is, an approach from the right leaning left in figure 1.1): one of the sub-tasks is to make inferences on a cropped image; Bayesian inference is used for this sub-task. Using the entire frame as Y_t is appropriate for a holistic Bayesian tracker (i.e. an approach from the left leaning right): there is a task; use all observations to infer a PDF on the output variables.

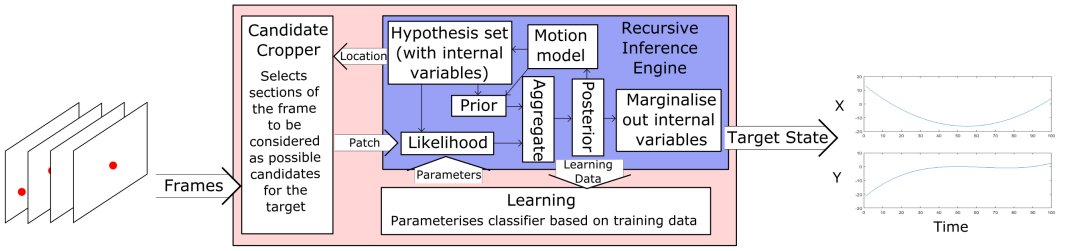
The second issue is the ad hoc model update. First note that both the observation and the motion models of the traditional BRE are time invariant. If $Y_i = Y_j$ and $X_i = X_j$ then we would expect $p(Y_i|X_i) = p(Y_j|X_j)$, and similarly for $p(X_i|X_{i-1})$. The mathematics does not allow for adaptation. Our solution will be the partitioning of X_i into a model component M and a state component S_t, so that the knowledge of M will be able to parameterize these models. While this may seem like an obvious fix that will simply describe the current adaptive trackers by lumping their ad hoc methods in M, we will prove its utility. Handling the observation model explicitly in M means that the posteriors are correctly normalised. Seeing the adaptive tracking problem as an inference across the joint (M, S_t) space will make us more aware of the approximations we are making. All this means that the final tracker will be more principled, and hence it will be easier to adapt to new situations. Not knowing what approximations are being made while using an ad hoc adaptive model means further ad hoc adjustments to find a solution that may or may not exist.

The issues we raise may seem trivial, however they characterise a different position on the approach spectrum. The strength in Bayesian methods is in their all-encompassing grasp on a problem. When we use a Bayesian tool for a subtask within a larger problem, we disempower the strength of its normalisation and rob the posteriors of their full meaning. This difference is illustrated in figure 1.3, which shows the difference between the characteristic architecture of current adaptive trackers and what we envision as a holistic Bayesian adaptive tracker. With this, it becomes clear that the above two issues that seem so peculiar for a holistic Bayesian tracker are just symptoms of a different model.

While current Bayesian trackers have success, any implementation of them that runs into challenges with the classifier must end up tweaking an ad hoc system. In a full Bayesian framework we can always test the validity of our approximations and the appropriateness of our priors. The fact that Bayesian systems are performing the optimal inference gives their results meaning, and ultimately makes them easier to debug. Thus, in striving



(A) The structure of our envisioned Bayesian tracker



(B) The general structure of most current Bayesian trackers

FIGURE 1.3: The difference between holistic and current Bayesian adaptive trackers. The holistic tracker (A) phrases the entire task as a single inference. Its internal structure mirrors the BRE equation (discussed in chapter 3); a set of hypotheses is generated by feeding the previous time instant's posterior through a motion model. Each of these hypotheses has its prior (using the motion model and previous posterior) and likelihood (using the current observations) calculated. These are then multiplied together and normalised across hypotheses to calculate the current time instant's posterior. Finally, any internal variables are marginalised out to leave only the relevant output variables. Most current Bayesian adaptive trackers can be reduced to the form of (B). Here the structure of the inference engine remains the same, but the input it receives is no longer the system's input and the likelihood function is updated in an ad hoc manner. Rather than encompassing the entire task, it is a functional block used inside another framework.

for a Bayesian framework that encompasses the entire task and not just the functional modules, we are not making an esoteric theoretical argument out of an irrelevant issue. Rather, we are striving to enable practical systems that are debuggable, maintainable, reasonable, and ultimately understandable.

We acknowledge that a holistic Bayesian tracker is not intrinsically better than current Bayesian trackers. All points along the approach spectrum have strengths and weaknesses. Our purpose in this discussion is to show that there is unexplored space to the left of current working trackers. We provide a more thorough justification for the above points in chapter 3, and derive a form of the BRE that fits in this space.

1.4 Maritime Surveillance

We will be testing our framework on maritime surveillance. While this may seem like a peculiarly specific choice to test a general framework, it is justified below.

The general tracking task is an ill-posed problem. The quantity of prior information a human brings into understanding a general scene is incredibly difficult to replicate. We want to see the framework's capability to learn and process commonalities in its input, but we lack the space to develop and test it on the same quantities of data received by our biological. The quality of inference a system can make is a function of the density of learning data in its problem space. When data is sparse, the prior must dominate behaviour. A design that works because of its prior demonstrates learning on the part of its designer, not the machines. By limiting the scope of the learning task, we select a problem that is tractable under the quantity of information we can give the system. We chose maritime surveillance as there is structure in the background off which a naive system cannot leverage.

Maritime surveillance is an important economic activity, and also a non-trivial object tracking task. There are existing systems (which are described in section 2.2), however research continues as the problem has not been fully solved.

We pick an instantiation of the maritime surveillance problem such that our framework can show its strengths without requiring too involved a formulation. That is: automatically initiating a multiple target tracker (MTT) on mostly small vessels against a variety of sea conditions from a static camera positioned high above the ocean. Our goal is to ratify our framework, so we will include enough detail into the problem to show how it can accommodate challenges, but will not waste effort on unnecessary features. We discuss this choice more in section 4.1. Our choices are guided by the desire to provide a use case as an example for the application of our framework.

We develop three modules in approaching this task. The first is an effective salience filter, which we test against current approaches. With this, we propose a principled way of combining salience filters. The second is an adaptive MTT, which demonstrates our framework's application to observation model learning. The third is a persistent tracker: a tracker that can be installed and whose performance will improve over time. In the case of maritime surveillance the biggest challenge is wave noise, so we focus on improving rejection of waves.

We draw attention to the difference between an adaptive tracker and a persistent tracker. An adaptive tracker's task is to follow a single target while maintaining an appropriate appearance model. The adaptive tracker uses the information from each frame to prepare it for subsequent frames. A persistent tracker's task is the long-term tracking problem. It follows a scene, tracking any relevant targets that enter and leave, improving as its deployment time increases. Each instantiated track of a persistent tracker is an adaptive tracker, and the persistent tracker uses the results of each track to improve its models

for subsequent tracks. The relationship between a persistent tracker and each adaptive track is the same as the relationship between an adaptive tracker and each frame.

While the focus of our work is on the contributions to adaptive tracking in the framework developed, we observe some deficits in the current maritime surveillance literature. Specifically: a principled way of combining salience filters; a salience filter that works well on the challenging data sequences we tackle; and a principled way of improving wave rejection while tracking (that is, the use of our framework). We list these points under our novel contributions despite the fact that our focus is on the general framework more than on maritime surveillance, as we did not find them in the current literature and believe they do add to the field.

1.5 Use in Modelling Human Interactions

In working with our framework we found it increasingly applicable to our everyday interactions. While this is probably true of any Bayesian work, we found that considering a separate model component M and state component S modelled social interactions particularly well. That people never say what they mean is cliché, and the use of Bayesian inference to decode this is a straightforward idea. However, we found the separate modelling of the questions 'What is the person trying to say?' and 'What can I infer about the person?' to be a pertinent description of how we handle social situations. We explore this in more depth, along with its links to the Turing test, and apply it to a specific social ritual in chapter 7.

1.6 Summary of Novel Contributions

In this section we summarise our novel contributions so that the reader may easily identify them in the body of the work. These are covered in more detail throughout the document, and will be summarised again at the end of this document with a more thorough justification. In our work we:

- Illustrate the inappropriateness of the BRE to adaptive tracking.
- Formulate and implement a holistic Bayesian adaptive tracking framework that incorporates model estimation into the PDF.
- Draw attention to the difference between an observation model in the form of a probability distribution, and one that is held in a distribution as a probabilistic variable itself.

- Emphasise and prove that considering the whole frame as observation (which is the more principled way to approach tracking) is equivalent to considering the contrast between the foreground observation and the background observation models for a bounding box.
- Develop a MTT particle filter that handles the multiple hypotheses in a novel manner and avoids the holding of joint solutions.
- Create a salience filter for maritime surveillance that is an improvement on the current state of the art.
- Highlight an appropriate measure of data fusion for maritime salience filters.
- Develop a persistent maritime tracker that improves at wave rejection in a principled manner.

1.7 Document Overview

The rest of this document proceeds as follows: chapter 2 covers the relevant literature to which we refer throughout our work. Chapter 3 contains the derivation of a framework that can achieve Bayesian adaptive and persistent multiple target tracking, which we will call SMAE (for Simultaneous Modelling and Estimation). Chapters 4 and 5 describe the application of this framework to the task of maritime surveillance. In chapter 4 we cover the maritime data set we will be using, and develop a salience filter to be used as a pre-filter for our tracking system. With these topics addressed, we focus on the application of SMAE to maritime surveillance in chapter 5. Having verified that SMAE is of practical use, we explore some of the more abstract applications of the framework in chapter 7. On account of these being outside of the central scope of this work, we include them as an epilogue after our concluding discussion in chapter 6.

A cursory reading of this work might give the impression that it is primarily about the development of a system for maritime surveillance, seeing chapter 5 as the primary focus. This is not the case. Our focus is on creating a framework for fully Bayesian adaptive tracking into which smaller modules can fit. Rather than the derivation being a stepping stone, and hence secondary to the maritime chapter, the maritime chapter is a ratification of and hence secondary to the derivation chapter. This has two implications that should be addressed before we enter the bulk of the document. Firstly, because the framework contains the bulk of our contribution, we devote sufficient time and space to it in the derivation chapter. The reader may find this chapter longer than the equivalent chapters in works where the implementation is the primary contribution. Secondly, a

reader who skims the derivation may feel underwhelmed by the maritime chapter; it is after all not our major contribution, but is included to prove that the framework is not just impractical theory.

Chapter 2

Literature Review

In this chapter we cover the relevant literature to which we refer in the rest of the work. Object tracking is a large field, so we focus in on the most relevant contributions. In section 2.1 we cover general Bayesian tracking: approaches to the challenging and in many ways ill-posed problem of tracking arbitrary objects in a free-form environment. In preparation for our application in chapters 4 and 5, we survey the literature pertaining to maritime surveillance in section 2.2. We have already mentioned Jaynes' work [2] and recommend it to any reader: it does not fit concisely into our sections and there is nothing directly to which we want to draw attention, however it is foundational to the thinking behind our work and we will reference it often.

2.1 Bayesian Tracking

We assume the reader is familiar with the Bayesian recursive estimator (BRE), and its instantiations as the Kalman filter for 'nice' systems and the particle filter as an approximation. We do cover the BRE briefly in section 3.2 as a lead up to our framework. We omit repetition of the discussion of the trackers by Pèrez et al. [3], Zhang et al. [4], Mei and Ling [5], Kwon and Lee [6] and Babenko et al. [7] presented in section 1.3. While there are many adaptive trackers that use Bayesian justifications for the tracking framework, we were unable to find any that include the adaptive observation model in that framework. The overview of Bayesian tracking in video presented by Dore et al. [8] is another useful resource, covering the theoretical foundations underpinning many of these trackers. While there have been many advances in many of the fields around machine vision through deep learning and specifically convolutional neural networks (CNN's), we will not cover these as they epitomise the right of the approach spectrum, and have little to do with the current work.

Multiple hypothesis tracking (MHT) was presented by Reid [9], and in many ways is a foundation for most Bayesian multi-object trackers. His proposed multiple object tracker is suited more to radar-like contacts than to video sequences, but it creates a powerful framework that many use. Upon each time instant, each measurement (i.e. co-ordinates of possible target) is assigned to either an existing object, a new object, or a false positive. The permutations of all these allocation history choices are held as different hypotheses. He presents several ways of keeping the branching tree of hypotheses manageable. The first option is a zero-scan algorithm: collapsing the distribution down to only one hypothesis at each time instant, either using the most likely hypothesis or a weighted sum of the hypotheses (JPDAF will be discussed below). The second option is multiple-scan algorithms: maintain the most likely hypotheses, and join hypotheses together if they are similar enough. This is tractable in the radar context, but becomes extremely tangled in adaptive tracking. The third option is simplifying the hypothesis matrix and initiating confirmed targets: looking for associations that are the same in all viable hypotheses and separating them into different clusters, thus only considering permutations among relevant targets. This is similar to how our implementation works. Another approach we have seen [10] is to limit the depth of any forks in the hypothesis tree. This implies that ambiguity in the underlying system should only last a set period of time, after which the most likely case should win out. The MHT uses "all available information such as density of unknown targets, density of false targets, probability of detection, and location uncertainty" [9]. If these values are unknown, or imprecisely known, the framework does not have a way to accommodate them. The framework we propose will encompass not only the model uncertainty in an adaptive tracker (i.e. a single object's appearance model), but also uncertainty relating to macro-variables like those mentioned that affect the whole task.

Fortmann et al. present the now widely-used joint probabilistic data association filter (JPDAF) [11] for Bayesian multiple target tracking in the context of radar. They consider the posterior probability of all the valid associations of contacts to targets. For each target's predicted location, they take the weighted average across all the associations. In this way they collapse the 'jointness' [1] of the distributions at each time instant. Considering the target's PDF as a union of the different permutations of associations is similar to the step in our framework where we consider a target's PDF as a union of the different sets of other visible targets. The approximation (collapsing the joint distribution) makes JPDAF a lightweight alternative to a more rigorous MHT (whose framework allows subsequent information to influence previous associations) in cases where the ambiguity in association is rare or limited in effect.

[1] That is, the co-dependence of targets' locations.

Kahn et al. [12] present a multi-object tracker that uses a Markov random field (MRF) defined on local clusters to modulate the sampling probability of the particles. These are considered as a joint random variable to avoid coalescence. Hess and Fern [13] use a similar approach to track American football players. The focus of their work is on discriminatively training the weights used for the re-sampling of the particle filter. In this way they avoid many of the 'jointness' problems associated with MTT particle filters (discussed in 3.3).

Koller et al. [14] present a multiple object tracker for road surveillance. This is similar to our problem in chapter 5: rigid vehicles moving along a 2D surface observed by a static camera. The main difference is the hostile nature of water as a background, in comparison to road surfaces. Background subtraction is a simple task for roads, and leads to far fewer 'clutter' objects. They model their objects by boundary contours and track the affine transformations on these contours. The focus of their work is on handling occlusions in a way that does not lead to errors in the affine parameter estimates. We find this noteworthy in that, although the task is similar, their approach is different. There is no adaptive modelling. Data association is done in an admittedly heuristic fashion, and while Kalman filters are used for shape estimation and motion tracking, there is a notable lack of Bayesian reasoning in occlusion. We believe that this is largely because tracking cars on a road is an easier task than tracking vessels (which have a greater variance in shape and size) on water (which exhibits more noise). We bring this up because our framework encompasses all of these challenges, updating models in a principled manner even while tracking occluding multiple objects.

The book by Stone et al. on Bayesian multiple target tracking [15] is an excellent reference. There are two topics we highlight. The first is tracking before detection, the essence of which is running a BRE across the entire state space rather than waiting for a cogent event to trigger initialisation. By doing this the need for a threshold is removed, and objects can be detected with much lower signal to noise ratios. They present convincing results in the detection of periscopes on a radar feed. The second topic is a multi-target version of the same idea, presented in section 4.4: a PDF is maintained for a two object, single-dimensional problem. This PDF is a two-dimensional one, where each dimension represents the location of one of the targets. We discuss this more in section 3.3.

Bloisi et al. [16] present a Bayesian multiple target multiple sensor[2] tracker. However, it is not an adaptive tracker; the visual classifier is trained offline to recognise boats using a cascade of Haar features, and this is used to detect boats. While the multi-target BRE seems similar to what we present, it represents the distribution as a Gaussian

[2] AIS (Automatic Identification System) data and surveillance cameras.

mixture model (GMM), which suggests that the targets are multiple modes in the same distribution (more on this in section 3.3). The handling of occlusions is done by tracking the merged entity and re-assigning the identities upon splitting. This is functional, and necessary with their approximation, but a step away from the Bayesian approach. We find that, while the system is successful, it does not explore the strengths of the Bayesian approach in both adaptive modelling and multi-target tracking.

Another trend worth mentioning is that of interactive multiple models (IMM) such as used by Blackman et al. [17] and Blom and Bloem [18]. IMM is an effective tool for objects that have several modes of motion, for example targets that can manoeuvre rapidly. While this has been shown to be useful, our focus is more on observation models, and our targets do not have rapid movement changes on the space-time scales we are at which we are observing. Hence we settle on a simple Gaussian motion model.

Many of the MTTs mentioned above are developed for radar [9, 11, 17]. Here the measurements are discrete contacts, which need to be allocated to different targets. This has several implications. Firstly, it means that there is no need or space for adaptive tracking. Secondly, data association becomes a well-posed task. In the case of visual object tracking, while we can associate on a pixel-by-pixel level it is often more practical to consider associations on a patch-by-patch basis. In doing so, handling the overlaps for different valid bounding boxes in a target's PDF can get involved. It is this intricacy that our derivation seeks to tackle in a principled manner. Radar formulations solve nice tractable forms of the problem, which make them good initial frameworks around which to build a visual object tracker.

Finally, we discuss some performance metrics used for tracking. It is common for trackers to use a measure for bounding-box overlap (such as Dice's coefficient, or the intersection-to-union ratio) or the distance between centroids to give a measure of how well a particular target matches the ground truth. Also common is to set a threshold on these measures to define hits and misses, and hence precision, recall and F-score for the results. These are good measures for single target trackers, but do not detect the failure modes associated with MTTs.

Smith et al. [19] cover a set of measures for the performance of MTTs. They define:

$$\text{FP} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\text{Number of false positives}}{\max(\text{number of visible targets},1)}$$

$$\text{FN} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\text{Number of missed targets}}{\max(\text{number of visible targets},1)}$$

$$\text{MO} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\sum_{\text{trackers on target}} (\text{number of targets best described by tracker} - 1)}{\max(\text{number of visible targets},1)}$$

$$\text{MT} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\sum_{\text{tracked targets}} (\text{number of tracks assigned to target} - 1)}{\max(\text{number of visible targets}, 1)}$$

$$\text{CD} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} |\text{number of targets - number of tracks}|$$

$$\text{FIT} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\text{number of tracks which are not the primary track for their target}}{\max(\text{number of visible targets}, 1)}$$

$$\text{FIO} = \frac{1}{\text{nFrames}} \sum_{\text{frames}} \frac{\text{number of targets tracked by a track other than their primary track}}{\max(\text{number of visible targets}, 1)}$$

$$\text{TP} = \frac{1}{\text{nTracks}} \sum_{\text{tracks}} \frac{\text{number of frames track is associated with its primary target}}{\text{number of frames track is visible}}$$

$$\text{OP} = \frac{1}{\text{nTargets}} \sum_{\text{targets}} \frac{\text{number of frames target is tracked by its primary tracker}}{\text{number of frames object is visible}}.$$

Here FP, FN, MO, MT and CD take into account errors within a particular frame — false positives, false negatives, multiple objects (covered by a particular track), multiple tracks (for a particular object) and counting errors; and FIT, FIO, TP and OP correspond to allocation errors — falsely identified track, falsely identified object, tracker purity and object purity. These last four assume that each track should be associated with a dominant target, and vice versa. Any track other than a target's dominant track or any target other than a track's dominant target is counted as an error. This can lead to peculiar situations, such as that in figure 2.1: under their measures a target that was tracked 51% of the time by one track and the rest by another track is outperformed by one which was tracked by one track 52% of the time, and then covered by 5 different tracks. We illustrate this point only with FIT errors, but the point is valid for the other three as well. We propose that the number of switches is a more relevant measure than percentage of time with most common track, and so use the SW measure defined in section 5.1.2.

Bernardin et al. [20] propose the multiple object tracking precision (MOTP) which describes the average distance target and response, and multiple object tracking accuracy (MOTA) which describes the frequency of false positive,false negatives, and mismatches. We chose not to use the former, as localisation is less important in our use case; or the later as it aggregates too many failure modes in one metric; but mention them here for completeness.

Much discussion has gone into the difference between generative trackers and discriminative trackers. Generative trackers focus on modelling the distribution from which the observations are drawn, whereas discriminative trackers focus on modelling the boundary between target and background in the considered feature space. This maps onto the approach spectrum (figure 1.1) well. Discriminative trackers exploit the fact that

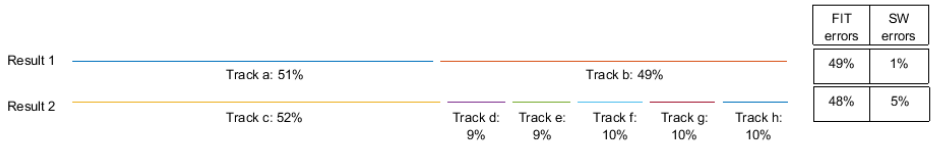


FIGURE 2.1: Two different tracking results and their FIT and SW error rates. Each row shows a possible tracking history for a target over 100 frames, with time as the horizontal axis. Each line segment represents a track that is associated with that target. Result 1 shows a target that was followed by one track for most of the time until it lost track, and was followed by another track for the rest of the time. Result 2 shows a similar situation, however once the initial track is lost, 5 different tracks are initiated and soon lost. The FIT counts any track that is not the dominant track for a target as an error, giving both of these similar ratings. The first result seems intuitively better, yet is rated worse according to FIT errors. Our measure counts the number of switches per frame, which matches our intuition for these results.

the only part of feature-space that is relevant to the tracking task is the region between foreground and background. Generative trackers, on the other hand, attempt to accurately estimate the full observation model. We would place discriminative trackers to the right of generative trackers.

Our approach will fall into the category of generative trackers. We draw attention to the difference between a distribution that observations are drawn from (i.e. $p(Y|X)$), and holding this observation distribution in a probabilistic variable (i.e. $p(M|X)$ where $M = p(Y|X)$). The first is a common trait of generative trackers, while the second is the focus of our work. It is the holding of the observation model as a probabilistic variable that enables us to extend the BRE to a framework applicable to adaptive tracking.

2.2 Maritime Tracking

In this section we discuss the relevant literature in maritime surveillance. We start in section 2.2.1 with an overview of the current state of the field. Section 2.2.2 covers the commonly used salience filters. In section 2.2.3, we look at the tracking frameworks that are commonly used. Section 2.2.4 discusses and provides examples of the data used by many papers, and section 2.2.5 addresses other relevant topics.

2.2.1 State of the Field

Moreira et al. present a good survey on maritime surveillance [21] that is fairly recent (2014). The main topics they discuss are horizon line detection, initial detection, and

vehicle tracking. The horizon line detection is not relevant to us, but the other two topics have a fair overlap with the use case we pursue. Much of the initial detection focuses on detecting salient patches to start tracking. Although they do not fit the form of a salience filter, this is a useful and relevant discussion which we include into section 2.2.2. Their section on vehicle tracking covers several common elements in a maritime tracker: Kalman filters, successive clustering, mean shift, template matching, histogram matching, and active contours. Most of the learning appears to be done in an offline fashion which, while reasonable for a facility as large as a harbour, does not explore Bayesian model updating. Several template-matching schemes are discussed, though none in as detailed a Bayesian update model as ours. The paper also discusses the use of FLIR cameras instead of sensing in the visible spectrum, discussing their insensitivity to lighting changes and white foam, but points out their high energy consumption and that 'they limit the quantity of features that can be extracted' [21]. We assume this is a comment on resolution. Many papers recommend the use of IR because the lack of temperature difference between the troughs and peaks of waves will lead to less noise, but the examples in section 2.2.4 show otherwise.

There are several working systems that have been deployed and tested in situ, namely ARGOS [10], MAAW [22], and DeMarine-DEKO [23].

ARGOS, presented by Bloisi and Iocchi [10], is an extended surveillance system for boat traffic monitoring in Venice. Their system does image segmentation via background subtraction, with improvements using optical flow and clustering, fed into a MHT system running on Kalman filters. Gupta et al. present the Maritime Activity Analysis Workbench (MAAW) [22]. Their work is focused on building a full system incorporating elements such as vehicle classification and behaviour interpretation. All learning is done in an extensive offline manner, and their use case seems to focus on boats in the harbour that fill an appreciable portion of the frame (it is unclear, as only one example image is present). The DeMarine-DEKO project by Saur et al. [23] focuses on synthetic aperture radar (SAR) images, which present very different problems and whose solutions do not have much impact on our problem. More recently, Bloisi et al. [16] presented a multiple target, multiple sensor system for maritime surveillance. This, however, uses a static system, in which all learning happens offline.

We list and describe these systems to show the level of complexity required to develop a workable solution. Our goal is not to outperform them; our goal is to tackle a challenging problem (to which the complexity in these systems and the ongoing research attests), and to prove that our framework provides a competitive solution. For these systems, the paradigm is to deploy a fully working system; our paradigm is to deploy a solution that has the capability to bootstrap itself to become a fully working solution. Thus our

focus is on producing passable initial results, and proven improvement through longer deployment. In this way our focus on Bayesian adaptive tracking presents a different approach that is absent in all of these papers.

There are many different modes of surveillance that are explored in the literature. They include 360° camera clusters on small boats [24], buoy-mounted systems [25], static mounted colour cameras [26], omni-cameras looking down the mast [27], satellite images [28], and pan-tilt-zoom cameras mounted on land [16] and on boats [29]. Each of these present different challenges and different advantages. We will focus on single static land-mounted cameras, as it is a common mode of surveillance and lends itself to our framework.

Many authors discuss the difficulties associated with maritime visual object tracking. Bachoo et al. [1] present a summary in figure 2.2.

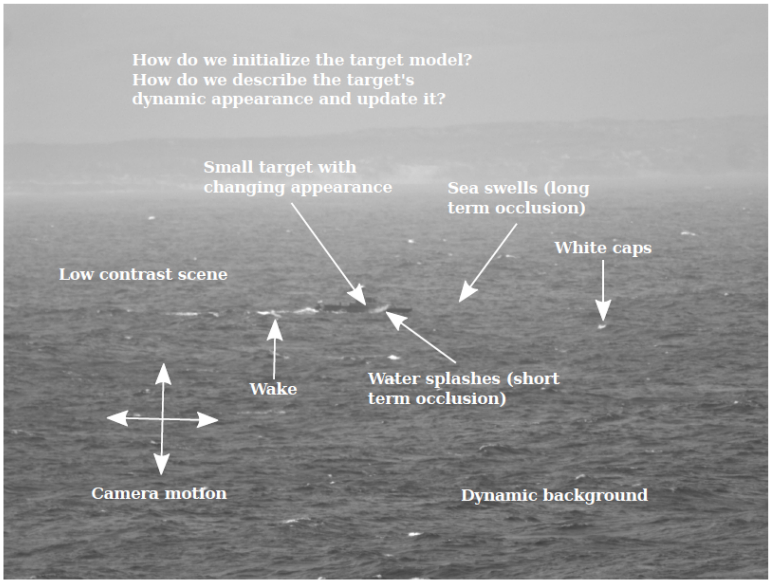


FIGURE 2.2: Summary of difficulties in maritime surveillance [1].

2.2.2 Salience Filters

Any system that automatically initialises needs a way to detect new targets — some method of detecting salient objects. In this section we cover the many techniques used to detect salient patches in a maritime sequence.

One of the two most common approaches to salience detection is building a history for each pixel, and marking pixels that do not fit their history. This history often takes the form of a probability distribution across pixel values, with low likelihood pixels presumed to be foreground. Alternatively, the history can be in the form of an

expected background, and deviation from this background image marks salience. Wang et al. [30] use an IIR filter on the input signal ($Bg_n = Bg_{n-1} + \alpha(Im_n - Bg_{n-1})$) to create a filtered image for background subtraction. They do not give a value for alpha. In such a system, there are trade-offs on the value of alpha. A large alpha value creates a short time-constant; this would include foreground that does not move quickly into the background. A small alpha value creates a long time-constant, making the system susceptible to sudden global changes (for example lighting changes, or camera movement). Inappropriate alpha choices can also lead to 'salience shadows' behind boats, where the background model has learned the foreground as its 'background' and flags the true background as different to its model (which is of foreground). Socek et al. [31] also use background subtraction learned through an IIR filter. Hu et al. [26] create a background image from a median value over 6 images, then salience is determined for each pixel as the smallest distance from the current pixel to its 9-neighbourhood in the background image. If a pixel is sufficiently different for a sufficiently long time period, the background will update again. Szpak and Tapamo [32] approximate each pixel's history as a Gaussian distribution. New values outside three standard deviations are marked as foreground, and put into a buffer. If they persist, they are considered background and are included into the distribution. Gupta et al. [22] perform background subtraction using a weighted average for past values, that favours medium-term historical values over recent values (which may be a salient object starting to appear) or long-term values (lighting conditions may have changed). Frost and Tapamo [33] use a kernel density estimate to model the history for each pixel taking n frames spaced s frames apart (by doing this, they avoid including slow-moving objects in the background without including a overly large number of history pixels). Bloisi and Iocchi [10] use a GMM with up to 7 modes to describe each pixel's background distribution. The histories are updated every 20 to 60 seconds depending on the lighting conditions. Robert et al. [34] group the image into macro-pixels, and test if the average value of the sub-pixel is sufficiently different from a single reference taken up to 3 minutes earlier.

The other common approach to salience detection builds its background model using the values in a single frame. This can be done either to set up a distribution, with unlikely values marked as salient, or as a background subtraction. Tao et al. [35] do a mean shift segmentation to detect salient regions. Liu et al. [27] use local color and edge orientation histograms. Patches are marked as containing salient objects if the distance (using the Bhattacharyya measure) between their histogram and the histogram of the area surrounding the patch is large enough. Smith and Teal [36] create a histogram for the average greyscale distribution of the sea; the tracker then measures the similarity between 32-by-32 pixel patches. If a patch's histogram gets a score of over 90% in its comparison to the sea's histogram (the paper does not define a comparison metric), then

the entire patch is marked as not salient. If the comparison scores less than 90%, 'the tile is further processed and classification is done on a pixel by pixel basis' [36]. Islam et al. [37] detect salient pixels by using a difference of Gaussians to detect pixels that are different from their surroundings. Selvi and Kumar [38] add the magnitude of the Sobel edge detector to the raw image, followed by a threshold that maximises inter-class variance while minimising intra-class variance. Saur et al. [23] use a combination of adaptive and absolute thresholds, which works on their SAR images. Pires et al. [39] approximate the whole sea frame using a Gaussian distribution, and look for outliers. Wei et al. [40] use iterative re-weighted least squares regression to fit a plane to the image before performing background subtraction. Teusch and Krüger [41] use several different markers to detect salient regions, then create a salience mask for the bounding box using several strategies. These involve a comparison to different regions: the horizontal mirror of the patch, the rest of the image, and the rest of the row. The considering of the entire frame's ocean as a static distribution (as in the second strategy) is a common trend, and works if the ocean is homogeneous. In many cases the model of the ocean fits better when considered by row (as in the third), as lighting and ocean conditions change as a function of distance, for which height in frame is a good proxy. This is advantageous over a static model for the ocean, and we will take the idea further in our salience filter, presented in section 4.3.2.

There are a few other approaches that do not fit into these two categories. Fefilatyev et al. [25] present a tracker mounted on a buoy. Because their camera is mounted so low, all boats to be detected are against the horizon line. Near the horizon line, water and sky have little texture, so they uses a thresholded colour gradient image on the image strip proximate to the horizon. Sanderson et al. [42] use a FFT on 32-by-32 pixel sub-windows, subtract from it an average frequency response (calculated from 10 sub-windows in the sequence), inverse FFT the difference, and finally stitch the patches together.

Several maritime trackers have no salience filters. Bachoo et al. [1] present a maritime tracker without a salience filter. Theirs is a first-generation single object tracker that uses a fairly off-the-shelf template tracker. They use a particle filter and deviation from template for the observation model, achieving good results. They only handle the STT case and require initialisation from a user. This explains their lack of a salience filter (which is frequently used to initialise tracks). Their straightforward approach solves the initialised-STT version of the problem well, whereas auto-initialised-MTT versions are very complex. This illustrates how large a component salience filters and initialisation is to the MTT task. Sullivan and Shah present another tracker that does not use a salience filter [43]. Instead, the tracker uses a maximum average correlation height (MACH) template built from extensive offline tracking, and looks for good matches

in the image using cross-correlation in the frequency domain. Bloisi and Iocchi [16] use a boosted cascade of Haar features, which are extensively trained offline. These classifiers are fast enough that they do not need a salience filter to limit the number of candidate targets tested. A surveillance system using this approach (learning different vessels offline and only searching for them) might be concerning, as any vessel that has a sufficiently peculiar shape may go undetected. In addition, our approach focuses on giving the tracker the minimum prior possible so that it can develop its own rich filter, thus excluding extensive training such as in these systems.

Bechar [44] presents a salience filter that is a combination of sub-filters (many trackers use multiple components, but Bechar's is the closest we could find to a Bayesian method). He sets

$$p(pix|obj) \propto 1 - \omega \times A \times B \times C \times D \tag{2.1}$$

where

- ω is a scaling variable,
- A $\propto$ Probability of drawing the pixel in question from the brightest class, assuming Gaussian observation distributions,
- B $\propto$ Probability of drawing the pixel in question from the bluest class, assuming Gaussian observation distributions,
- C $\propto$ is a measure of how spread out the colour is in the frame (higher spread means the color is less likely to be boat), and
- D is a measure of dynamic texture (correlation between neighbouring pixels).

The components are set up as probabilities; however, combining them in a 'noisy OR' like this is ad hoc. We will discuss a more appropriate method in section 4.3.1.

There are several post-processing steps that are used on the salience filter response to suppress noise. Small connected components are often ignored [10, 30, 33, 36, 40]. Frost and Tapamo [33] also use a variable-size threshold dependent on the y-value to perform a perspective-independent threshold on connected components. Bloisi and Iocchi [10] use clustering of sparse optical flow results to separate objects whose images form one connected component in the image (which also suppresses wakes). Morphological operators are frequently used [22, 31, 32, 40] to treat the image for salt-and-pepper noise. Wei et al. [40] use a structuring element whose size dependents on the y-position, in order to take perspective into account. Saur et al. [23] use the Hough transform to filter for boats whose outline is dominated by two long lines (which is relevant for SAR images).

Selecting for salient patches in which the bottom is a long straight line may work for large boats on calm seas, but we do not believe it would work well in our conditions. Frost and Tapamo [33] use motion persistence (area must be salient for a while). This implies a strong prior on $p(boat|speed)$, which we feel is unjustified.

A final point worth mentioning is brought up by Moreira et al. [21]. The success of a salience filter or initial vessel detector is dependent on the application. Different conditions will favour different filters, and so it is important to see the filters in light of the sequences on which they are run. It would have been preferable to include sample images from each paper in this section. However, that would have undermined the portrayal of the common structure in the literature, so we instead examine sample images in section 2.2.4.

2.2.3 Trackers

In this section we describe the commonly used trackers. There are three relevant topics: feature sets, classifiers, and trackers. We review these in turn.

There is a large variety of features used. We first list them before drawing together the common threads. Gupta et al. [22] use the feature set {position, velocity, Hu's image moments (up to 4th order), category of nearest object, distance to nearest object, bearing of nearest object}. Sanderson et al. [45] use {Hu's image moment, PCA of multiple templates, local features via the 2D Hadamard transform}. Feineigle et al. [46] use SIFT features (which is appropriate for the large vessels in their use case). Teutsch and Krüger [41] use a set of 342 features that include information from {Hu's moments, statistical image properties (variance, contrast, entropy), gradient features, local binary patterns}. To describe the set fully would take too much space. From this set they use a greedy algorithm to select the best features using linear discriminant analysis. Wei et al. [40] also use an extensive set of features including Gabor wavelet co-efficients, colour histograms, histograms of edge points, edge orientation, edge length, shape moments, angular area histogram, Fourier descriptors, and boundary statistical features (mean value, standard deviation and histograms of curvatures)[3]. Alfadda [47] uses GIST (a spatial/scale array of Gabor-like filters), HOG and DeCAF (deep convolutional activation feature) and a collection of other features (LBP, LBPHF, line features, SSIM, texton histograms, geometric probability map and SIFT). Mattyus [28] and Bloisi et al. [16] use Haar features. Bousetoune and Morris [48] use CNNs trained on large data sets (ImageNet) stripped of their last layer to generate features from the second last

[3]Some of these seem vague, and the original paper does not offer much of a description.

layer. Selvi and Kumar [38] use aspect ratio, compactness (perimeter^2/area), convexity (area/area-of-convex-hull), first seven image moments, statistical measures of grey value (mean, variance, moments, and entropy), and wavelet-based features (as described in their paper). Bloisi and Iocchi [10] do not use features, rather tracking clusters of salience (with optical flow to separate collisions). Liu et al. [27] use no features, but instead track the best ellipse of salience (switching between colour spaces favouring the most discriminating one). Bachoo et al. [1] and Hu et al. [26] both use the raw pixel values as the features to be used for tracking with a template.

The main trend in this list is the lack of a dominant technique. While many of the common tracking features are present (Haar, HOG, SIFT etc.), they form a minority. Most of the repeated features are more structural features of the image, such as Hu's moment or gray-scale entropy. We believe this is because waves are noisy, and any feature that focuses on detail will find a patch of water displaying similar detail. The larger structure of the target, on the other hand, is more discriminative. This is also pertinent in our case, as we focus on smaller boats and so will usually have less space for detail-based features.

Once features have been decided upon, they need to be used. Usually some sort of classifier is used to decide which candidate best fits the object or class of interest. Teutsch and Krüger [41] use two SVM stages to classify targets of interest; the first distinguishes between objects and clutter, and the second distinguishes suspicious boats from other objects. Sanderson et al. [45] test the functionality of both a single Gaussian and a GMM approximation of the training examples to classify new samples. Selvi and Kumar [38] propose using either a KNN or a SVM classifier (although neither is actually implemented in their paper). Mattyus [28] uses an adaBoost framework, Wei et al. [40] use a decision forest, and Bloisi et al. [16] use a boosted cascade of classifiers.

While the general tracking problem sees much activity in the form of authors testing new classifier frameworks, the classifier is seldom the focus in maritime surveillance papers. We list these examples to show that, although some may have Bayesian foundations, none approach the adaptive tracking with the thorough framework we develop.

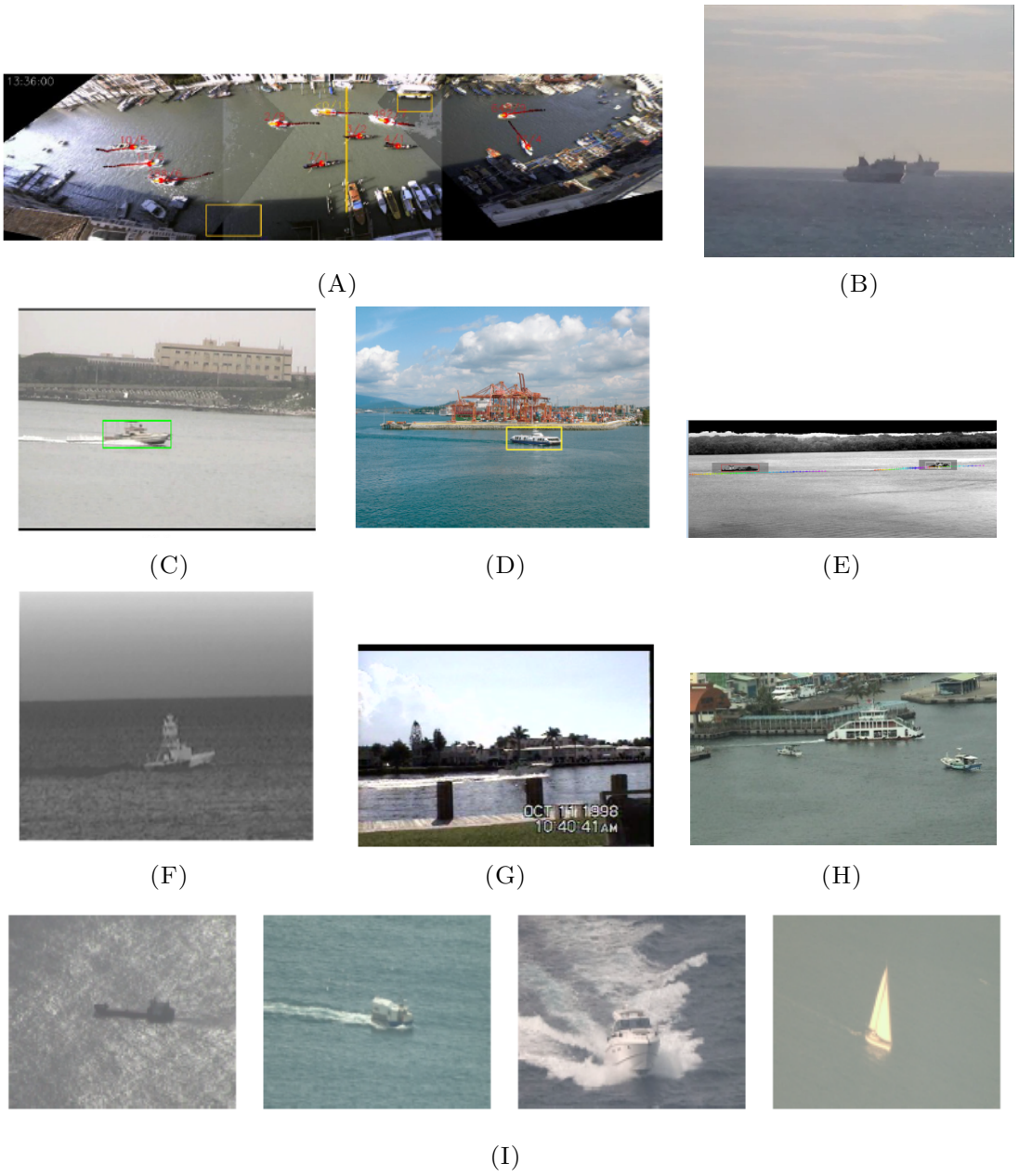
Several tracking frameworks occur repeatedly, namely: mean-shift [27, 29, 35], the Kalman filter [10, 27, 49], MHT [10, 25], and active contours [32, 33]. A more thorough analysis is given by Moreira et al. [21].

2.2.4 Data Sets

For any task in machine vision the performance of the system is intricately linked with the data set on which it is tested. This is especially true in maritime surveillance, as the condition of the ocean has a large variance and can change the background from a nearly homogeneous sheet to cluttered, noisy turmoil. The choice of boat size relative to the frame also has a large effect on the challenge to the system. In this section we present sample images and describe the testing data set from the referenced papers, so that the reader has an idea of the instances of the problem tackled by current systems. We display a representative sample of other authors' data sets that the relative difficulty of our data set, presented in section 4.2, may be appreciated.

Our description of the sample frames and the availability of data may sound critical. This is not to cast judgement on the authors: we understand the space limitations that go with journal publication and the difficulties of presenting representative samples of video sets. We adopt candour so the reader will understand the challenges in perceiving the level of difficulty of data that is standard in the field.

Bloisi and Iocchi [10] tested their system (ARGOS) extensively in situ; a sample frame is shown in figure 2.3(A). Alfadda [47] and Bloisi et al. [16] use the MarDCT data sets available from `www.dis.uniroma1.it/~labrococo/MAR/`, which has 20 sequences for detection, 4774 cropped images for classification, and 9 sequences for tracking. A sample frame from one of the tracking sequences is shown in figure 2.3(B). Hu et al. [26] do not say where their data comes from, but it is implied that it is self-generated. Most of the experiments occur on only one sequence. A sample frame is shown in figure 2.3(C). Sullivan and Shah [43] recorded their own footage, and used sequences available online (`www.uscg.mil`, `www.bbcmotiongallery.com`, and `www.nvmc.uscg.gov`). Several sample frames are shown (e.g. figure 2.3(D)), yet the quantity of data from each source is not mentioned. Gupta et al. [22] analysed two days of video of from the Potomac River in Washington, with 1578 tracked objects. The only example image given is figure 2.3(E). Tao et al. [35] test their segmentation algorithm on six frames from a single sequence (sample shown in figure 2.3(F)). Socek et al. [31] refer to a number of scenes, but only show one (sample image in figure 2.3(G)). No quantitative results are given. Wang et al. [30] test their algorithm with 3 sequences (sample frame in figure 2.3(H)). Bechar et al. [44] show 4 'shots' (shown in figure 2.3(I)); it is unclear whether this is a cropping from the frame (they seem to have narrow FOVs) or the whole frame. They claim to have run their algorithm on 'a dozen of realistic sequences' [44], but no quantitative results are shown.

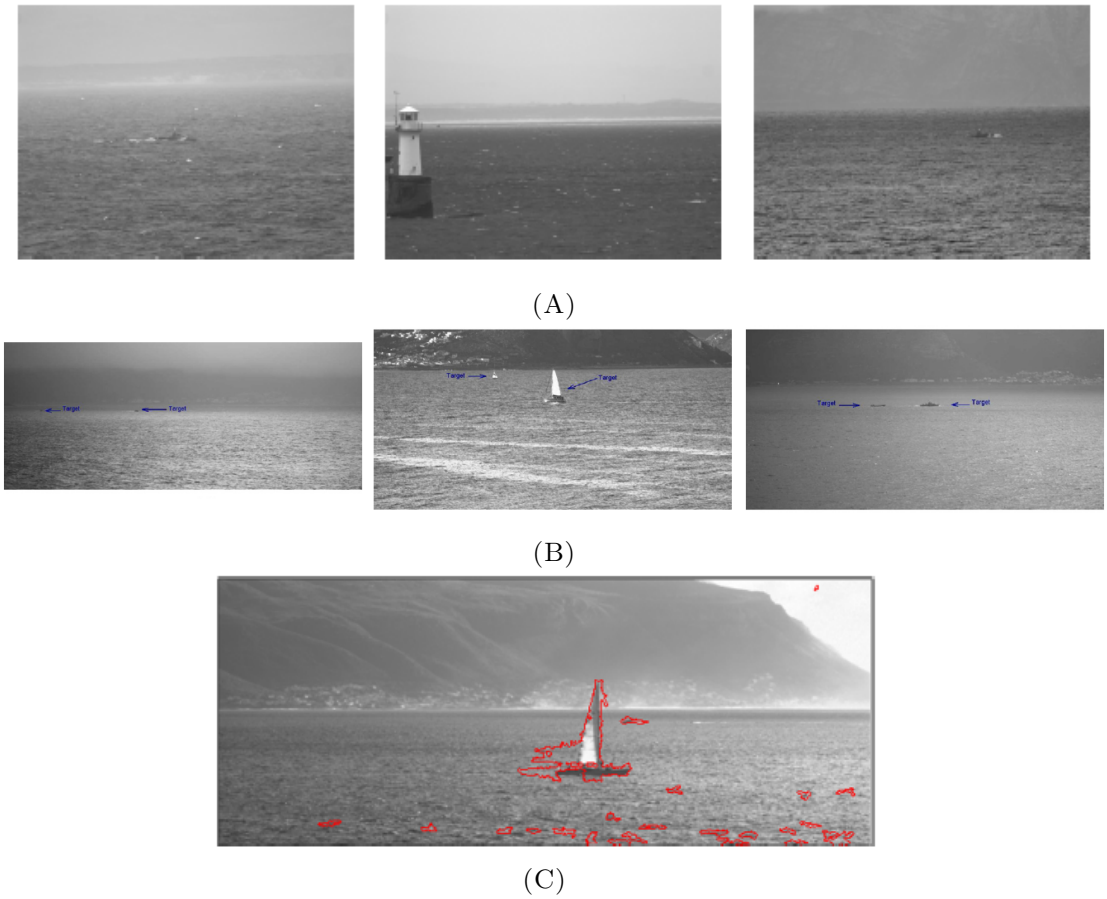


(A) (B) (C) (D) (E) (F) (G) (H) (I)

FIGURE 2.3: Sample images from current literature review: Bloisi and Iocchi [10] (A), the MarDCT data set used by Alfadda [47] and Bloisi et al. [16] (B), Hu et al. [26] (C), Sullivan and Shah [43] (D), Gupta et al. [22] (E), Tao et al. [35] (F), Socek et al. [31] (G), Wang et al. [30] (H), Bechar et al. [44] (I) (it is unclear whether these are frames, or cropped from frames).

The following papers' data sets were captured on the South African coast, which has harsher sea conditions, as opposed to conditions on the Mediterranean and in harbours. Bachoo et al. [1] use three data sequences, sample frames of which are shown in figure 2.4(A). Szpak and Tapamo [32] test their algorithm on four sequences, giving sample frames in pseudo-colour for all of them; the three greyscale images they give are shown in figure 2.4(B). Frost and Tapamo [33] use a set of ten maritime sequences obtained from the Council of Science and Industrial Research (South Africa). The only

sample frame is shown in figure 2.4(C). These sequences have more challenging ocean conditions, and smaller targets relative to the frame.



(A)

(B)

(C)

FIGURE 2.4: Sample images from current literature review (captured in South Africa): Bachoo et al. [1] (A), Szpak and Tapamo [32] (B), Frost and Tapamo [33] (C).

Many systems use IR cameras. Wei et al. [40] tested their surveillance system on data provided by Lockheed Martin, including visible and IR sequences (sample frames in figure 2.5(A) and figure 2.5(B) respectively). The extent and exact nature of all the data is not provided. Islam et al. [37] test their system on 132 standard and IR camera images from 'army night vision lab'[sic]. Sample images are shown in figure 2.5(C) and figure 2.5(D). Robert et al. [34] test their system on two visual and IR sequences, samples of which are shown in figure 2.5(E) and 2.5(F). Krüger and Orlov [50] claim their system is tested on a large set of IR sequences captured in the North Sea. Several sample frames are given (e.g. figure 2.5(G) and figure 2.5(H)), but no quantitative results are given. Pires et al. [39] test their system on four sequences from different IR cameras, giving a thorough description of each sequence (example frames in figure 2.5(I) and figure 2.5(J)). Teutsch and Krüger [41] test their algorithm with 19 sequences (sample frame in figure 2.5(K)). However, not all sequences are described, and no reference is given.

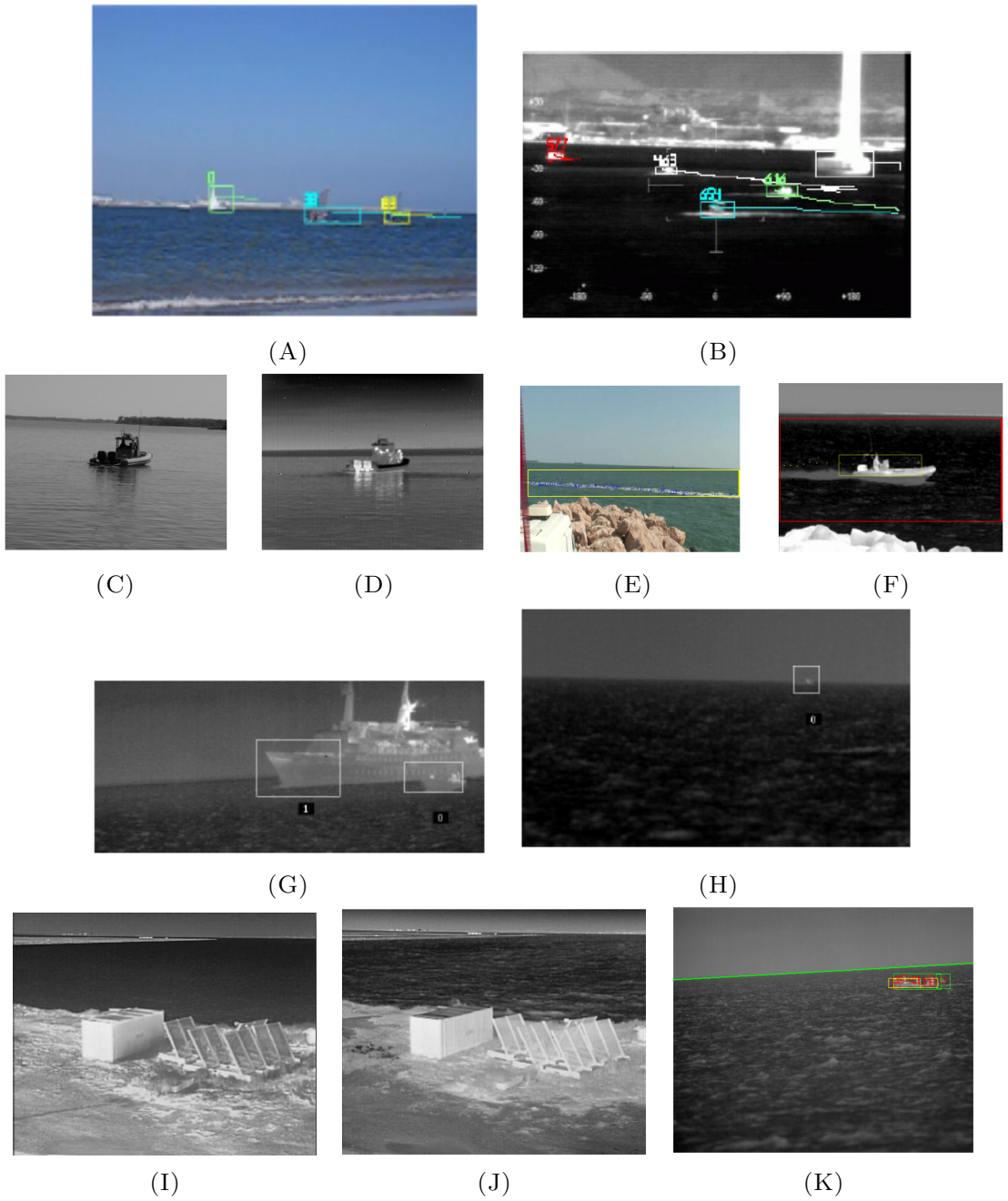


FIGURE 2.5: Sample images from current literature review (papers including IR): Wei et al. [40] (A) and (B) (visible spectrum and IR respectively), Islam et al. [37] (C) and (D) (visible spectrum and IR respectively), Robert et al. [34] (E) and (F) (visible spectrum and IR respectively), Krüger and Orlov [50] (G) and (H) (both IR, showing different target size), Pires et al. [39] (I) and (J) (both IR, showing different conditions), Teutsch and Krüger [41] (K).

We do not include data samples from the classification papers, as they contain pre-cropped boat images, and similarly for the papers with different surveillance modes (e.g. buoy-mounted, satellite images, etc.). We list the above trackers to show the trends in the current maritime literature: commonly the focus is on large vessels and easy sea conditions; algorithms are tested on a few sequences; often data is not available, and only

a few sample frames are shown; results are usually focused on qualitative illustrations of a working system, rather than on quantitative results (if any are given). We believe this is because of the nature of maritime surveillance. All authors have their own use case in mind, and their own setup to construct. Any reader will likely be solving a slightly different problem; quantitative results are not necessarily relevant to that problem. Value can be found in qualitative results showing the sorts of data a technique works on and the sorts of results it can give. This is very different to the standard VOT community, which is built around standardised data sets and comparative quantitative results.

2.2.5 Other Pertinent Topics

Finally, we address several other points raised in the literature that we find pertinent.

The tracker developed by Fefilatyev et al. [25] is bouy-mounted, and deals with large amounts of camera movement. A fair portion of their paper addresses horizon detection, for which they assume that most of the frame is divided into two sets of pixels (sky and water). They test the validity of this assumption for each frame, and if it is false (due either to extreme camera pitch or a boat filling the frame) they discard the frame. We find this noteworthy for two reasons: having a concrete statement of your assumptions is useful, and building checks for those assumptions is useful as well. Also, in dealing with edge cases, such as when assumptions are wrong, the choice of an algorithm's response must be a function of the task at hand. The assumption that boats are small makes the algorithm weak in detecting large boats. In our case, this is safe as it is the smaller boats that are more likely to be missed by human operators.

Szpak and Tapamo [32] justify the choice of grey-scale sequences over colour sequences: long range cameras are usually gray-scale due to the resolution-colour trade off, and it also avoids blue channel saturation.

Lastly, the frequent comment that IR cameras produce more homogeneous ocean colouring should be viewed in light of figure 2.5: many IR images still display wave noise, and have an intensity gradient as the ocean tends towards the horizon.

Chapter 3

Derivation of Mathematical Framework

We now move on to the main contribution of this work. In this chapter we derive a Bayesian framework that approaches the adaptive MTT problem in a holistic manner, framing the entire task in its inference. Because it is easier to follow the derivation with a concrete problem, we use a small synthetic problem to visualise the different variables and processes.

We start off in section 3.1, by introducing the synthetic problem with which we walk through the derivation. In section 3.2 we consider the single target version of this problem, discussing the common Bayesian approaches and introducing our framework. Section 3.3 extends this framework to the adaptive MTT case and discusses the problems with traditional approaches. We address the difference between adaptive MTT and a persistent tracker (and our framework's encompassing of both) in section 3.4. Finally, we present the changes that would be necessary to extend the tracker from the sample problem to a visual object tracking (VOT) problem in section 3.5.

3.1 Synthetic Problem

For our synthetic problem we consider a one-dimensional tracking task where the observations are a line of 40 scalar-valued pixels Y_t ranging from 0 to 1, and the targets are single pixel aberrations on the background. For readability, we will show these values as a chromatic change instead of gray-scale. We draw background pixels from a Gaussian distribution centred on zero, and targets will be drawn from a Gaussian distribution centred on μ (which is a parameter of the object). All observation distributions will

have a standard deviation of 0.1, and will be clipped to the range $[0, 1]$. The targets will have a state with a single variable x (the location of the aberration) which will follow a Gaussian random walk with a standard deviation of 1 pixels.

Figure 3.1 shows a sample instance of the problem. The horizontal axis represents the dimension the targets are being tracked across, and the vertical axis represents time (increasing towards the bottom of the image). Low values are shown with blue pixels, high values with red pixels. Three targets move through the scene with μ values of 0.15, 0.7 and 0.5 from left to right at the start of the sequence. The image on the left represents the ground truth; the image on the right is the actual observations.

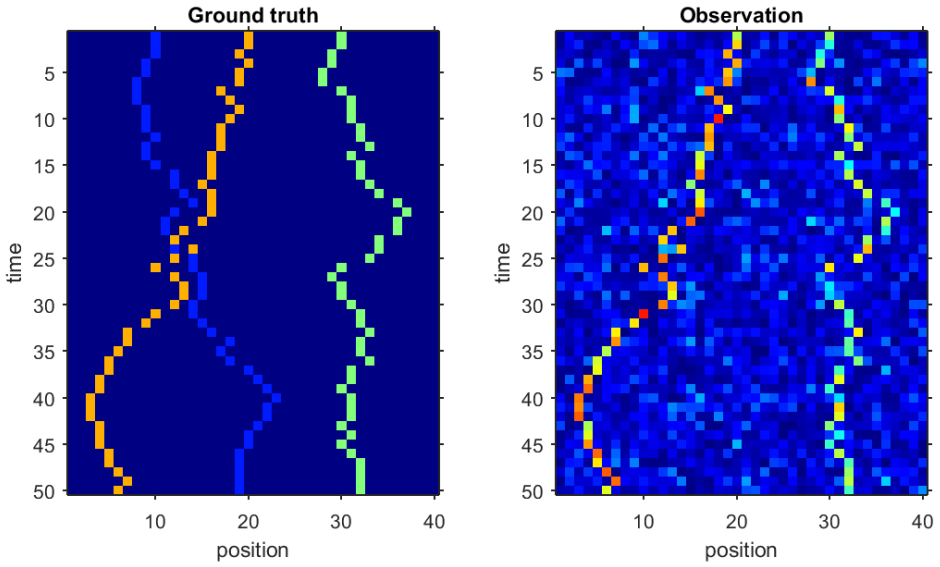


FIGURE 3.1: Sample instantiation of the synthetic one-dimensional adaptive tracking problem.

3.2 Adaptive Single Target Tracking

We first consider a single target tracking (STT) version of the problem. A fairly common approach to adaptive tracking is the use of a BRE (through its approximations the Kalman filter and the particle filter). The BRE poses the tracking problem as a hidden Markov model (HMM) in which the underlying state (which we will call X_t) evolves in a stochastic manner that is only dependent on the current value (i.e. it has the Markov property), and the estimator has access only to observation variables (which we will call Y_t) that are dependent on the state. We will call the distribution $p(X_t|X_{t-1})$ the transition or motion model, and the distribution $p(Y_t|X_t)$ the observation model (which

is called a likelihood if seen as a function of X_t). Figure 3.2 shows a graphical model of a HMM.

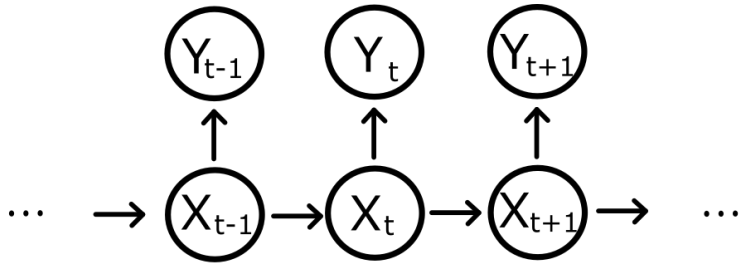


FIGURE 3.2: Bayesian network for a HMM. At each time instant, the current state X_t is dependent only on the previous time instant X_{t-1}. The state is not observable, so all reasoning about the system state needs to be done through the observable variables Y_t that are dependent on X_t.

Application of Bayes' rule reduces the posterior to

$$p(X_t|Y_{1:t}) = \frac{p(Y_t|X_t, Y_{1:t-1}) \int p(X_t|X_{t-1}, Y_{1:t-1}) p(X_{t-1}|Y_{1:t-1}) dX_{t-1}}{p(Y_t|Y_{1:t-1})}. \tag{3.1}$$

This is often broken up into a prediction step (the contents of the integral) and an update step (scaling the prediction by the likelihood, followed by normalisation). Because the system has the Markov property, the history $Y_{1:t-1}$ holds no extra information and can be dropped from the motion and observation models, giving

$$p(X_t|Y_{1:t}) = \frac{p(Y_t|X_t) \int p(X_t|X_{t-1}) p(X_{t-1}|Y_{1:t-1}) dX_{t-1}}{p(Y_t|Y_{1:t-1})}. \tag{3.2}$$

The major problems with this formulation stem from its assumption that the motion and observation models are known a priori. Adaptive trackers adapt in order to overcome a lack of this exact information at initialisation time. This problem can be seen in the equation: $p(X_t|X_{t-1})$ and $p(Y_t|X_t)$ are time-independent. Given $X_{i-1:i} = X_{j-1:j}$, $Y_i = Y_j$ and $i \neq j$, we should have $p(X_i|X_{i-1}) = p(X_j|X_{j-1})$ and $p(Y_i|X_i) = p(Y_j|X_j)$. However, this would not be true for an adaptive tracker. The history of observations $Y_{1:t-1}$ is still relevant, and hence the Markov property is not appropriate.

The obvious fix is to include more into our state X_t until the Markov property is met. In order to do so, we partition the state X_t into two components M and S_t. The quantity S_t contains all the time-varying properties we traditionally see as the state, and M contains

all the information we have learnt in previous frames[1]. The quantity M will get more intricate later, but for now it will only hold a distribution on the HMM models. After substituting[2], our equation becomes

$$p(M, S_t|Y_{1:t}) = \frac{p(Y_t|M, S_t) \int p(S_t|M, S_{t-1}) p(M, S_{t-1}|Y_{1:t-1}) dX_{t-1}}{p(Y_t|Y_{1:t-1})}. \tag{3.3}$$

This has much the same structure as equation 3.2, except that now we are maintaining a PDF across the joint (M,S_t) space. The model M is now given for both the motion and the observation models, thus it can modulate them. Even though M is not a time-varying property (as indicated by its lack of subscript), each point in the joint space has specific observation/motion models attached to it. There are two ways of thinking about this process. The first (more accurate) way is to think of each model having its own inference engine that runs in parallel to the rest, calculating its own likelihood and prediction. The normalisation then occurs across all models together. The second is to imagine the PDF moving through this joint space, with different motion and observation models becoming appropriate as it moves. This second approach is tempting, yet misleading. Even though the weight of the PDF may move in the M dimensions, the normal movement (i.e. the diffusion that happens during the prediction step) in the inference occurs only along the S_t dimensions. Any transfer of mass in the PDF that occurs along the M dimensions is a result of normalisation. A sample iteration of this inference is illustrated in figure 3.3.

While we use M to parameterise a simple model space, the system could accommodate convoluted spaces that select between any number of different observation models: all that is required is that these models make predictions for $p(X_t|M, X_{t-1})$ and $p(Y_t|M, X_t)$[3]. To name this framework, we note that the key factor that differentiates it from the BRE is the principled incorporation of model adjustment, and so we call it simultaneous modelling and estimation, or SMAE.

Our sample problem is small enough that we can maintain a decent discretised version of the PDF for the joint (M,S_t) space[4]. Our state is defined by a single variable x, and the model by the centre of the observation distribution μ. We will assume a known motion model for simplicity (however, the parameters for it could easily be included in M). We define each pixel p's observation to be Y_t^p and assume independent pixels (a

[1] It is tempting to put in a subscript M_t, however the model is not a time-varying property; it is our information about M that changes with further observations. Hence $p(M|Y_{1:t})$ is a function of time, but M is not.

[2] Recognising that $p(M|M, S_{1:t-1})$=1; that is, the motion model of X_t is static in all dimensions associated with M (by definition).

[3] One could say that all these functions are already in our M domain, and that we have set their prior to zero.

[4] We are still considering the case in which we assume there is only one target.

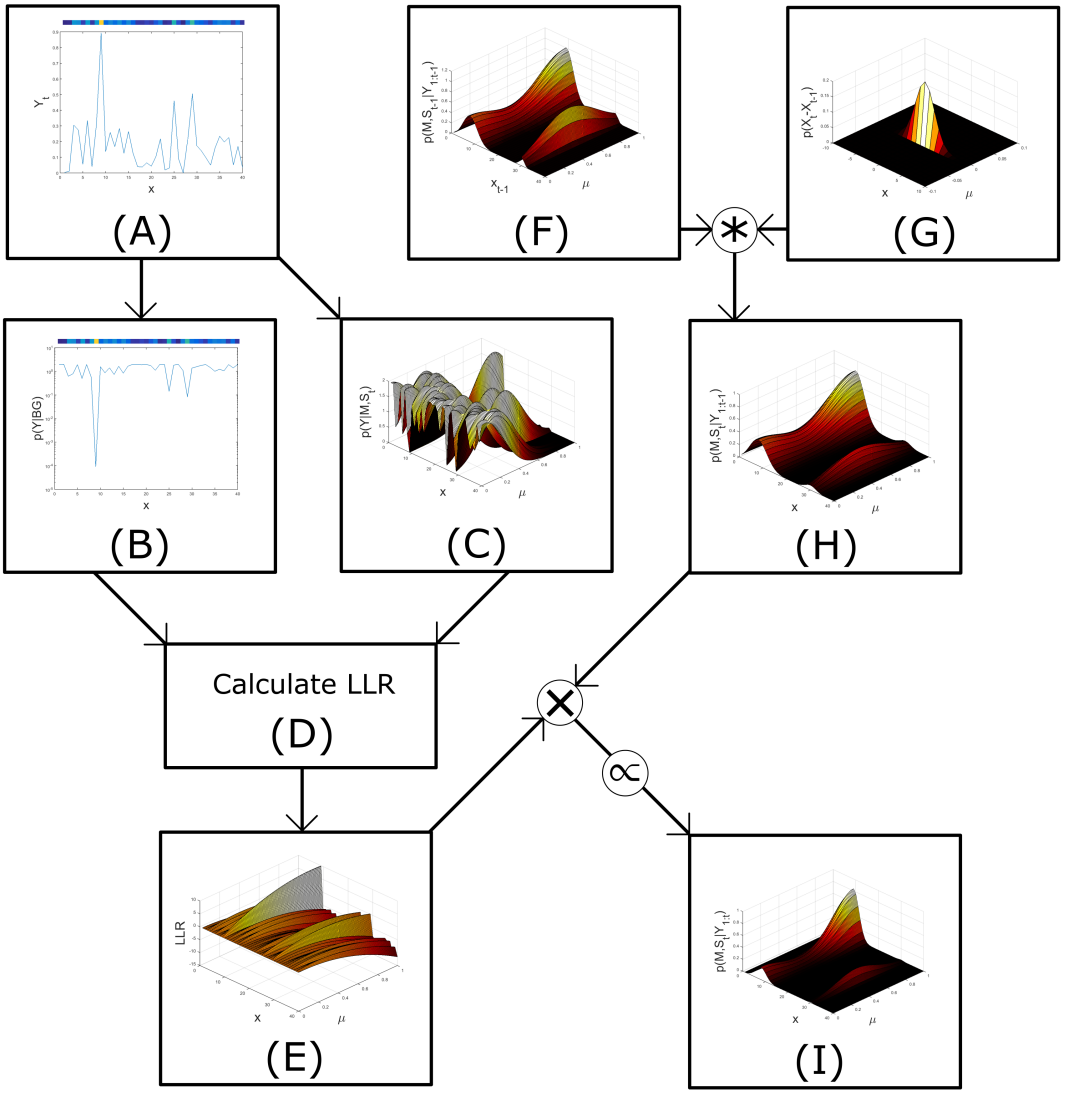


FIGURE 3.3: A single inference step for the synthetic problem. The observations (Y_t) for a particular time t are shown in (A). Each pixel (Y_t^p) in the observation is used to calculate background likelihoods ($p(Y_t|Bg_p)$) in (B) and foreground likelihoods ($p(Y_t^p|M, S_t)$) in (C). As an object affects only the pixel value at its location, the foreground likelihood is composed of slices for each pixel location, defining how well each model predicts the observed pixel. We look at this more in figures 3.4, 3.12 and 3.13, and define it as the update function. The calculation of the log likelihood ratio is done in (D). Each slice in (C) is divided by the appropriate value from (B), and the quotient's log is shown in (E). This forms the observation half of equation 3.2. The posterior for the previous time instance ($p(M, S_{t-1}|Y_{1:t-1})$) is shown in (F). This is updated by convolution with the motion model (G) to produce the prediction ($p(M, S_t|Y_{1:t-1})$) (H). The motion model never moves PDF mass from one M value to another, as all the time-varying properties are in S_t. Each M plane can define its own motion model. The prediction and the observation model are combined according to equation 3.2: using point-by-point multiplication, with the result normalised across the entire (M,S_t) plane, to create our posterior $p(M, S_t|Y_{1:t})$) (I). Our notation in this diagram assumes (E) is not in log-space; we do not include the converting of (E) and (H) to a common space for diagram complexity sake.

false but common assumption), and get the observation model

$$p(Y_t|M, S_t) = \prod_{Y_t^p} p(Y_t^p|M, S_t). \tag{3.4}$$

This simply states that the likelihood of the entire observation is the product of the likelihoods for each pixel. Because an object with state $S_t = x$ only makes predictions about the pixel at location x (that is Y_t^x), this becomes

$$p(Y_t|M, S_t) = K.\frac{p(Y_t^x|M, S_t)}{p(Y_t^x|Bg_x)}. \tag{3.5}$$

Here Bg_p is the event that pixel p was drawn from the background distribution, and

$$K = \prod_{Y_t^p} p(Y_t^p|Bg_p). \tag{3.6}$$

This constant K is independent of M and S_t, and hence disappears during the normalisation. Thus it can be ignored. Dropping K and putting in the likelihoods, we get

$$p(Y_t|M, S_t) = \frac{1}{\sigma_{obs}\sqrt{2\pi}} e^{\frac{-1}{2(\sigma_{obs})^2}\left((\mu - Y_t^x)^2 - (Y_t^x)^2\right)}. \tag{3.7}$$

Here, the likelihood of observing the entire observation has been reduced to a function of only the relevant pixel (Y_t^x). This decomposition (which is equally valid in standard tracking problems) illuminates the peculiarity in the trend of only considering the bounding box (or in our case, the pixel) as Y_t. In figure 1.3, we presented the difference between a holistic Bayesian tracker and the current Bayesian approaches. It is more principled to include all pixels in Y_t, and is mathematically equivalent to using a likelihood ratio for only pixels in the bounding box. Even if a LLR is not desired, setting a uniform (and hence irrelevant after normalisation) background model will lead to using only the bounding box as Y_t. Hence we can switch our thinking to the more principled approach (fitting the whole task into a Bayesian inference engine) and benefit from it without having to change our practice. Thus the use of a bounding box is justified by our framework.

Putting this together with a motion model $p(S_t = x_1|M, S_{t-1} = x_2) \sim \mathcal{N}(x_1 - x_2, 1^2)$, and a quantised grid of μ with a step of 0.05, we can run SMAE on our synthetic problem under the temporary assumption that there is only one target. Each time-step is a three-step process:

1) diffuse $p(S_{t-1}, M|Y_{1:t-1})$ according to the motion model to get $p(S_t, M|Y_{1:t-1})$,

2) evaluate the likelihood for each point in state space, and

3) normalise across the entire (M,S_t) space.

For step 2, each location x has a PDF for μ and an observation Y_t^x. We can show the update (that is, the combining of prior and likelihood) for each considered state S_t as multiplication with an update function. Figure 3.4 shows this update function (in μ) as a function of Y_t^x. Here the Y_t^x axis represents the input, and the slice defined by setting Y_t^x constant is the update function to use for a given Y_t^x. The update function (which is a likelihood ratio) is shown on a log scale, as the graph is dominated by high μ, high Y_t^x values. As one would expect, the update function for a given observation peaks at a μ value equal to that observation. However, the value of that peak changes dramatically. The more difficult it is to explain an observation as background, the more weight it lends to the models that could explain it. Considering that the normalisation is done across the entire (M,S_t) plane, it is easy for the peaks associated with smaller μ values to become negligible. Under the assumption that there is exactly one target in the scene, the contest between hypothesised targets is as much about which observations the background struggles to explain as it is about which observations fit the foreground models[5]. This is as a result of considering the entire observation as Y and is desirable behaviour: if an adaptive tracker is allowed to pick its own observations, it can pick only those values that fit its model. This would lead to confirmation bias. The denominator of the LLR (which was a result of our normalising E out of the likelihood) is what pushes the tracker towards more pertinent potential targets.

Figure 3.5 shows results for SMAE on the synthetic problem[6]. The top row shows the observations given to the estimator, the ground truth locations for all targets in the data, and the posterior PDF marginalised across μ and x. The data was generated with three targets, yet the tracker assumes there is only one. One might hope that the PDF would be split between the three targets, but the tracker's PDF is focused entirely on the object with the highest μ. The distribution on the μ can be seen to concentrate on the correct value for this target as time progresses. Rows two to four show the prior (row 2), likelihood ratio (row 3), and posterior (row 4) for 5 different instances in time. Notice that the prior gets more certain of μ as time progresses, but maintains its breadth in x (in accordance with the motion model). The update (i.e. likelihood ratio) is dominated by the response at the location of the primary target. The magnitude of the peak depends greatly on the observed value: this is seen in the value of the peak in

[5] This is relevant in the next section.

[6] A note on colour palettes: we use different colour maps for different common plots to help differentiate their meaning. LLRs shown as a function of the observation pixel and model value will use the Autumn (red to yellow) color map, distributions over the (M,S_t) space for a particular time will use Hot (black through red to white), and images where time is an axis will use Parula (Blue through green to orange).

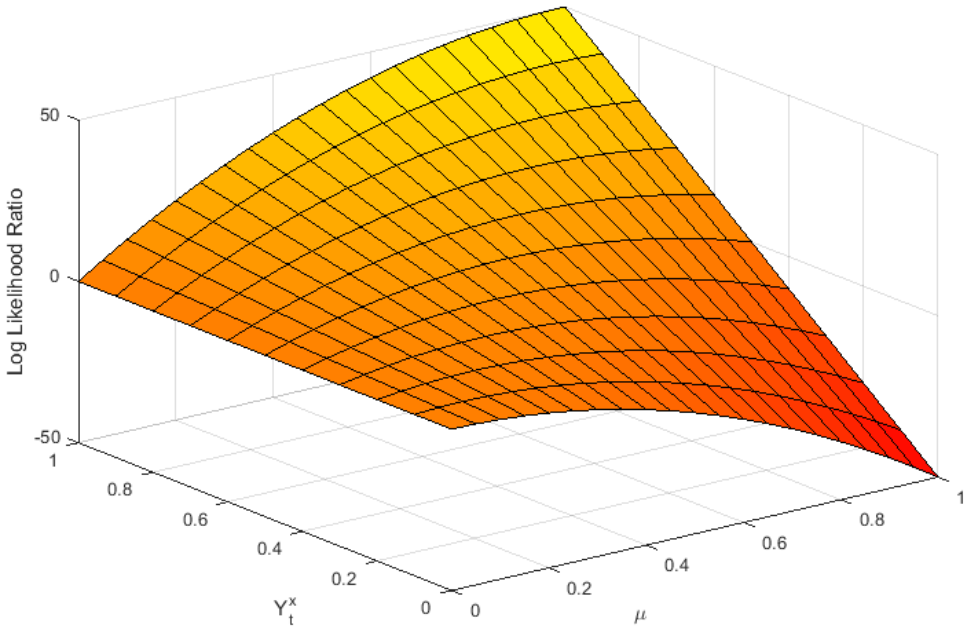


FIGURE 3.4: Model update function for synthetic problem. For a given observation Y_t^x, the distribution on μ at x will be multiplied in a point-by-point fashion with the intersection of the plane defined by Y_t^x and the above surface.

row 3 changing by six orders of magnitude, depending on its location in μ. Secondary peaks in the update can be seen at the location of the secondary target in frames 1 and 43, due to the combination of low observation values for the primary target and high values on the secondary target. Unfortunately, these are some of the very few frames in which the secondary target is not negligible, and the posterior focuses on the primary target due to division by a tiny $p(Y_t^x|Bg_x)$. The posterior follows the primary target well, and we can see the uncertainty in M decrease.

We observe that the system effectively tracks and models the single most pertinent target in the scenario. Now we move on to multiple target tracking.

3.3 Adaptive Multiple Target Tracking

Our derivation moves through an initial approach to multiple target tracking(MTT) in section 3.3.1, a particle-like filter formulation in section 3.3.2, and finally covers complications due to overlaps in section 3.3.3.

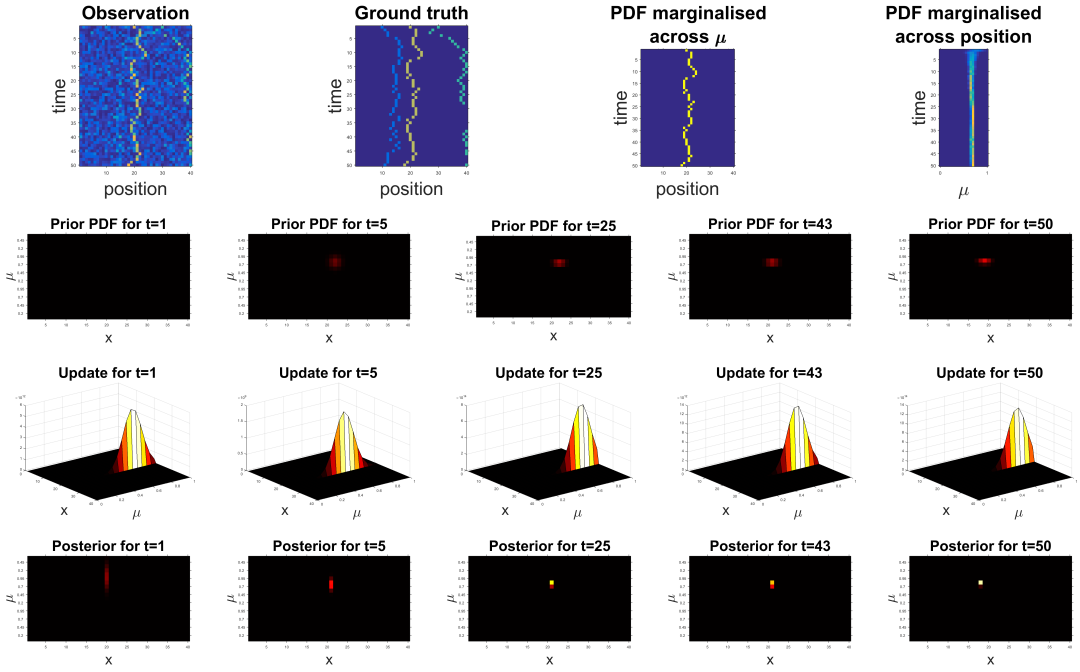


FIGURE 3.5: Results for the synthetic problem as STT. The top row shows observations given to the estimator, the ground truth locations for all targets in the data, and the posterior PDF marginalised across μ and x. Rows two to four show the prior (row 2), likelihood ratio (row 3), and posterior (row 4) for 5 different instances in time.

3.3.1 Initial Approach

The PDF in the above analysis held the posterior distribution across the (M,S_t) plane under the assumption that there was exactly one target in the scene. While one might hope that multiple targets in the scene would lead to multiple modes in the distribution, the results for the synthetic problem show that the denominator in the LLR can lead to the most pertinent[7] target's response drowning out any others. This is made worse when targets interact or pass close to each other, which may lead to coalescence (where two tracks tracking two different targets merge, and follow the favoured of the two tracks).

To handle the situation correctly we need to note that our current (M,S_t) plane is no longer appropriate as the domain for the PDF. Stone et al. [15] present a similar example to ours (except without the model parameter μ), in which they model two targets moving in a one-dimensional space. Because they knew the number of objects, they could maintain the joint PDF across them. It would be possible to model the joint PDF for two objects in our scenario, creating a 4D PDF across the variables M', S'_t ,M'', and S''_t. Displaying a 4D PDF is non-trivial, so we adapt our definition of $p(M, S_t)$. We no longer consider it as the state of a specific object, but rather let it represent the probability that there is an object with model M at state S_t. Thus our PDF for a

[7] The target most difficult to explain as background.

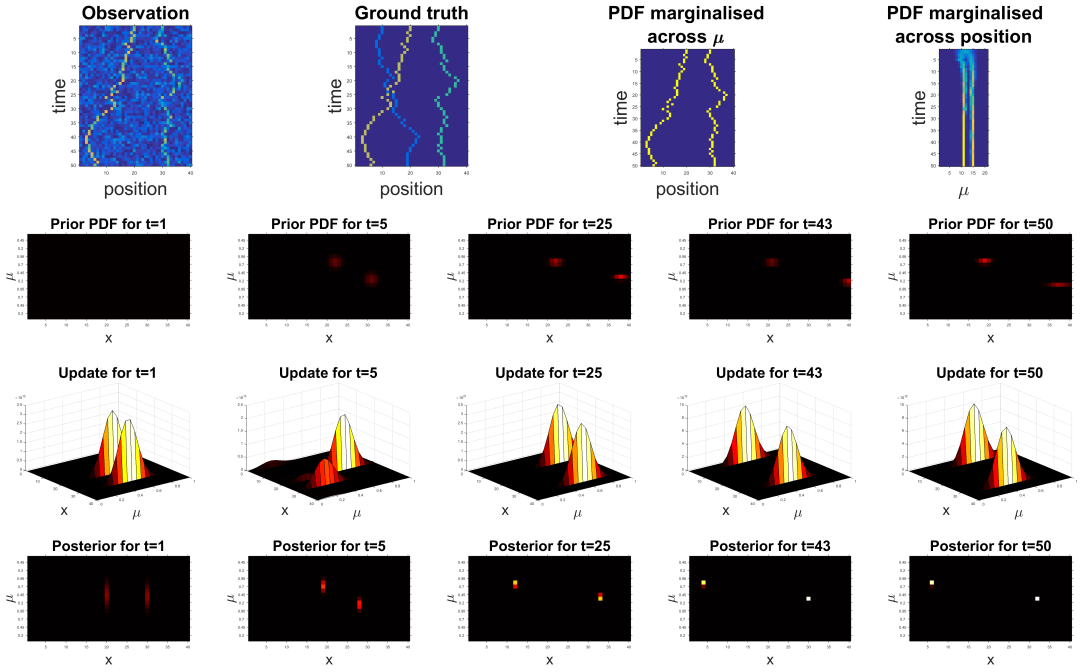


FIGURE 3.6: Results for the synthetic problem assuming that exactly two targets are visible. The top row shows observations given to the estimator, the ground truth locations for all targets in the data, and the $p(M, S_t)$ as defined in the text, marginalised across μ and x. Rows two to four show the prior (row 2), likelihood ratio (row 3), and posterior (row 4) for 5 different instances in time (the 4-dimensional PDF has been collapsed into a 2D representation as described in the text).

point on the (M, S_t) plane at time t can be seen as the sum of the PDFs for the joint two-target distribution with target 1 at the point, marginalised across all possibilities for target 2, and the mirror scenario (which will have an identical value). That is,

$$p(M, S_t) = \int p(M, S_t, M'', S_t'')dM''dS_t'' + \int p(M', S_t', M, S_t)dM'dS_t'. \tag{3.8}$$

These results are shown in figure 3.6. Rather than attempt to display the 4D joint PDF across two target (M, S_t) space, we display the $p(M, S_t)$ as defined above.

The plots are the similar to figure 3.5. Many of the same patterns can be seen in these graphs: the posterior localises the position S_t well, except now it manages to follow two targets. The M components start off unknown, but localise as more observations are taken into account. Any targets beyond the assumed number are ignored. The update's magnitude varies dramatically, based on the denominator of the likelihood ratio. The main difference is that the update can now have multiple modes.

The shortfall of this approach is twofold: maintaining a joint PDF becomes exponentially more complex as more targets are considered, and this still assumes a set, known number of targets in view. A Bayesian approach to MTT with an unknown number of targets would be to consider the domain of X_t as the union of all 2N-dimensioned subspaces

describing the joints across N targets. This, however, is not feasible. We will follow a less expensive approach with our newly defined $p(M, S_t|Y_{1:t})$. Rather than answering the question 'How likely is a target with model M at S_t?' under the assumption that only one target is present, we ask this question without that assumption. The Bayesian inference is similar. Our prediction will still take the previous step's posterior, and feed it through a motion model to get a prior. Our likelihood is still easily defined by what model M predicts as observations[8]; it is just the new normalisation that needs to be addressed.

3.3.2 The Use of Particle-like Filters

Managing this will be easier if we hold the distribution in a discretised form. As our solution for the non-synthetic problems will use particle-like filters, it is natural to switch to one now.

The formulation will not be a traditional particle filter, in which a dense set of samples approximate a non-parametric PDF. Unfortunately, the dimensionality of M in our final application will rule out dense sampling, and so we will use a hybridised system to store our PDF. Our particles $\hat{X}_t$ will be discrete in S_t, but maintain a distribution in M as shown in figure 3.7 (this is possible as each update is multiplication with a Gaussian). In this way, dimensions associated with S_t are modelled using a particle-like filter, but the problems associated with dimensionality are avoided by handling M parametrically. Our filter will also have dramatically fewer particles than traditional particle filters, but this will be discussed below. The name 'particle-like' filter is distracting, hence we will refer to our formulation simply as a particle filter in the text. However, we draw attention to the fact that our filter is not a traditional particle filter lest the differences distract the reader.

It will be necessary to define relations between the different particles, to determine which can and cannot co-exist. To do this, we group particles in clusters C_t [9] such that particles in a cluster must co-exist, and we keep a record of which pairs of clusters cannot co-exist. If we now define the set of visible targets at time t as A_t, we can see our posterior (now defined on particles $\hat{X}$ rather than on (M,S_t)) as marginalised across all allowed sets of particles A_t^j:

$$p(\hat{X}_t|Y_{1:t}) = \sum_{A_t^j} p(\hat{X}_t|A_t^j, Y_{1:t}) p(A_t^j|Y_{1:t}). \tag{3.9}$$

[8] We address the intricacies involved in overlapping templates in section 3.3.3.
[9] As per the third option in MHT mentioned in section 2.1

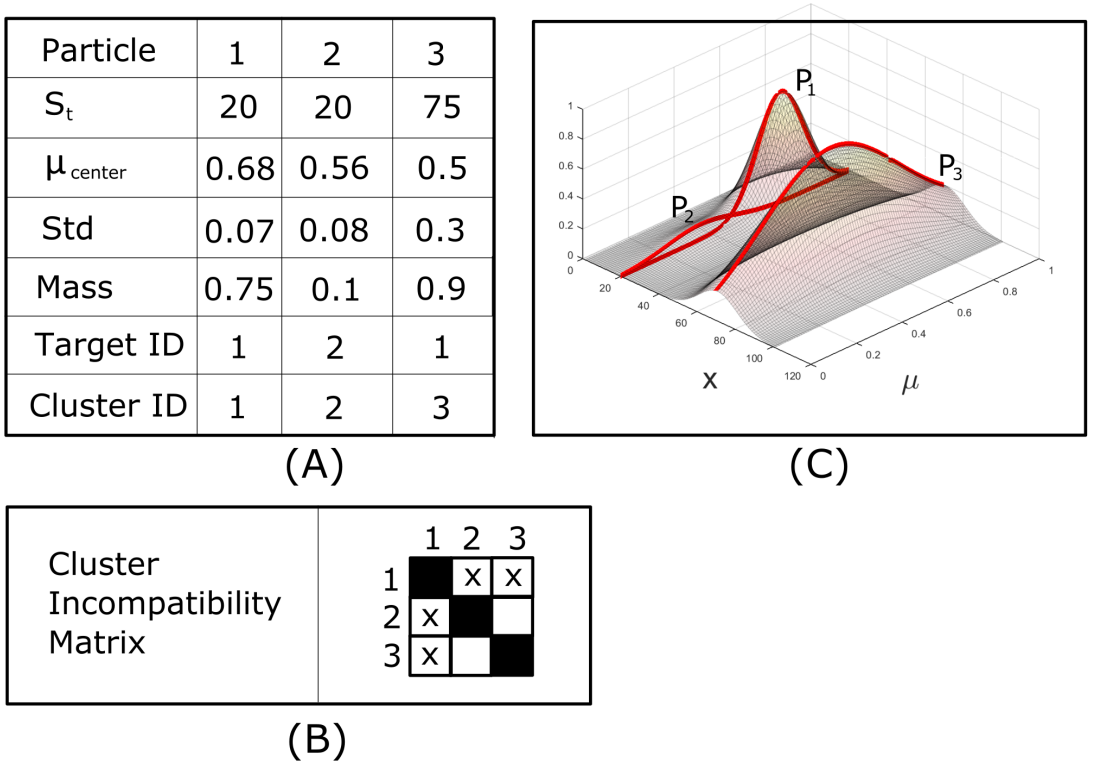


FIGURE 3.7: The illustration of MTT for a particular frame of the synthetic problem using a particle filter. The left (A) shows the summary of the particles in question. The right (B) shows an illustration of these particles. The lower (C) shows the compatibility matrix: Particles 1 and 3 are incompatible because they claim the same history (the particle in the prior cannot go to both, so they must compete against one another). Particles 1 and 2 are marked as incompatible because they overlap; this case is explored more thoroughly in the text, and in figure 3.10 which explores the need for clusters. For now, each particle is its own cluster.

Thus the posterior on a particle is the sum of its posteriors given each of the possible visible sets A_t^j, scaled by the posterior on each set. Rearranging the second factor and noticing that the first factor simply tests if the particle $\hat{X}_t$ is in A_t^j gives us

$$p(\hat{X}_t|Y_{1:t}) = \sum_{A_t^j} bool(\hat{X}_t \in A_t^j) \frac{p(Y_t|A_t)p(A_t^j|Y_{1:t-1})}{p(Y_t|Y_{1:t-1})}. \tag{3.10}$$

Here the denominator is the numerator integrated across A_t^j; the system is normalised across all the valid subsets.

Because the pixels are independent, we can split our likelihood:

$$p(Y_t|A_t^j) = \prod_{Y_i^p} p(Y_i^p|A_t^j). \tag{3.11}$$

Particles that deal with the same pixels will either be incompatible (and hence will not co-exist in a valid A_t) or be in the same cluster, thus the pixels in the likelihood can be separated by cluster. Our clusters will not interfere with each other (we join interacting clusters into one new cluster), hence we can split our priors as well:

$$p(\hat{X}_t|Y_{1:t}) = \sum_{A_t^j} bool(\hat{X}_t \in A_t^j) \frac{\prod_{C_y^m \in A_t^j} \left(\left(\prod_{Y_i^p \in C_t^m} \frac{p(Y_i^p|C_t^m)}{p(Y_i^p|Bg_p)} \right) p(C_t^m|Y_{1:t-1}) \right)}{p(Y_t|Y_{1:t-1})}. \tag{3.12}$$

Here again we have ignored the constant $K = \prod_{Y_i^p} p(Y_i^p|Bg_p)$, which will disappear in the normalisation. We split the equation semantically to become

$$p(\hat{X}_t|Y_{1:t}) = \frac{\sum_{A_t^j} bool(\hat{X}_t \in A_t^j) \prod_{C_t^m \in A_t^j} p(C_t^j)}{p(Y_t|Y_{1:t-1})}, \tag{3.13}$$

with

$$p(C_t^m) = \left(\prod_{y_i^p \in C_t^m} \frac{p(Y_i^p|C_t^m)}{p(Y_i^p|Bg_p)} \right) . p(C_t^m|Y_{1:t-1}). \tag{3.14}$$

We use $p(C_t^m)$ as shorthand to represent the parts of the equation that relate to a cluster, rather than with a more accurate but obfuscating variable name. In equation 3.14, each cluster receives a score that is its contribution to any posterior in which it is used. This contribution is the product of its prior, and the LLR on each pixel for which it makes a prediction. In equation 3.13 we iterate over the valid subsets of clusters. For each subset, we add the product of the cluster's contribution into an accumulator for each of the particles used. Once all the subsets have been considered, we normalise the accumulators to get the posteriors for each particle. The normalisation constant can be written as

$$p(Y_t|Y_{1:t-1}) = \sum_{A_t^j} p(Y_t|A_t^j, Y_{1:t-1}) p(A_t^j|Y_{1:t-1}). \tag{3.15}$$

We have already split these terms across clusters in equation 3.12, making

$$p(Y_t|Y_{1:t-1}) = \sum_{A_t^j} \prod_{C_t^m \in A_t^j} p(C_t^m). \tag{3.16}$$

.

Thus our normalisation constant can be calculated while iterating through subsets by adding to a normalisation accumulator for each subset A_t^j. This process is illustrated in figure 3.8.

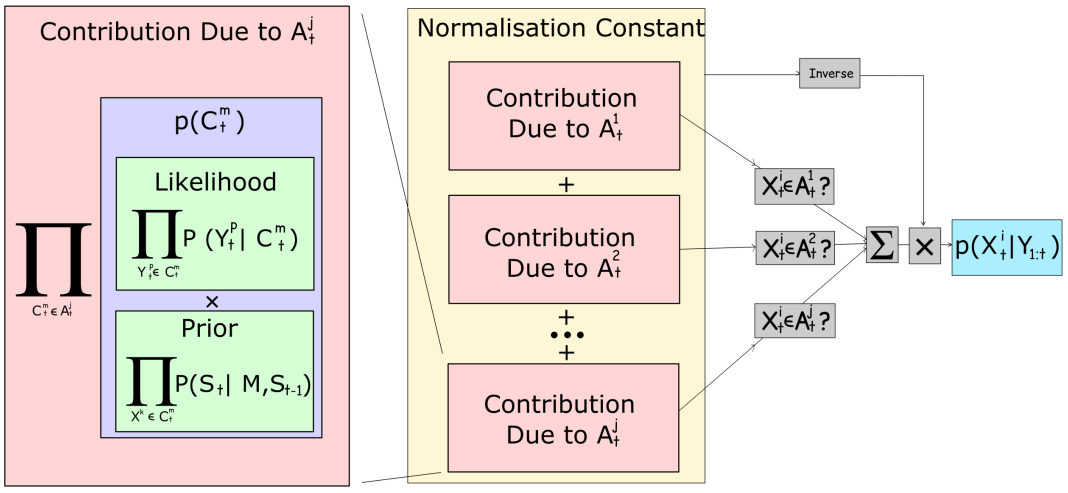


FIGURE 3.8: Multiple-target particle filter algorithm. Each cluster's contribution ($p(C_t^m)$) is calculated as the product of its prior (the product of all its constituent particles' priors) and its likelihood (the product of likelihoods for all the pixels it addresses). The viable subset of clusters A_t^j is iterated over, and each is assigned a contribution equal to the product of all the clusters it includes. The final posterior for particle X_t^i is the sum of contributions for the A_t^j that contain X_t^i, normalised by the sum of all A_t^j contributions.

3.3.3 Inference During Collisions

For clusters that are single particles, or for pixels in a cluster that only one particle addresses, the above process is simple. However, we need to define what happens to the likelihood $p(Y_t^p|C_t^m)$ if multiple particles in a cluster address it. To handle this we name those particles in C_t^m that are completing over a specific pixel $\hat{X}^k$ ($k \in \mathbb{N}$). For each particle, we define the likelihood that its observation model would be expressed in the absence of all other particles as α_k (for now this is always 1, but we will address observation models in which it is not). Now for multiple particles trying to express their observation models, we give each particle a success probability (β_k) proportional to its α_k, and the background a probability (β_0) proportional to $\prod(1 - \alpha_k)$:

$$\beta_k \propto \alpha_k, \quad k \in \mathbb{N}_0, \tag{3.17}$$

with

$$\alpha_0 = \prod(1 - \alpha_k), \quad k \in \mathbb{N}, \tag{3.18}$$

and

$$\sum \beta_k = 1, \qquad k \in \mathbb{N}_0. \tag{3.19}$$

Defining D_k to be the event that the actual observation originated from $\hat{X}^k$, we get

$$p(Y_t^p|C_t^m) = \sum p(Y_t^p|D_k, C_t^m)p(D_k|C_t^m). \tag{3.20}$$

Here the only relevant part of C_t^m, once D_k is known, is M_k. Thus

$$p(Y_t^p|C_t^m) = \sum p(Y_t^p|D_k, M_k)p(D_k|C_t^m). \tag{3.21}$$

Given that this is inside an inference, we obtain

$$p(C_t^m|Y_t^p) \propto \sum p(Y_t^p|D_k, M_k)p(D_k|C_t^m)p(C_t^m). \tag{3.22}$$

We are going to be maintaining the distribution on each M_k in its own particle. Hence, we make the approximation that the M_k are independent from each other; this allows us to maintain atomic particles without the construction of complex joint models across the particles within a cluster. With this approximation, we can manage each M_k in C_t^m separately (that is, marginalising across all $M_{k'}$; $k' \neq k$):

$$p(M_k|Y_i^p) \propto p(Y_t^p|D_k, M_k)\beta_k p(M_k) + \sum_{k' \neq k} \int p(Y_t^p|D_{k'}, M_{k'})\beta_{k'} p(M_{k'}) dM_{k'} p(M_k). \tag{3.23}$$

Equation 3.23 has two parts. The first term is the prior on M_k multiplied by the likelihood, scaled by β_k. This (without the β_k) is the distribution on M_k we would have had as a posterior, had only $\hat{X}_k$ been at position p. The other term is the sum of the mass in the equivalent first terms for all the other k-values, which is constant in M_k, multiplied by the prior on M_k. The end effect is that each M_k becomes a weighted sum of its distribution, updated by the observation (that is, the leading term of equation 3.23) and its prior (the summation in equation 3.23). The weightings are determined by which particle's model best fits the observation. The closer a particle $\hat{X}_k$ is to being the dominant explanation of an observation, the more weight the updated distribution gets. If another particle $\hat{X}_{k'}$ meets the observation better, then the weight for M_k's calculation goes to the prior. We make one last approximation, and that is to fit a single Gaussian to this sum of Gaussian distributions. Usually this is an inaccurate approximation to make. In our case, however, if the means are far apart it implies that M_k does not fit the observation well, and thus the dominant weight is likely to be in the prior (the second term of 3.23). The further apart the two Gaussians we are grouping are (the condition for an inferior single Gaussian approximation), the more the updated distribution will become negligible (i.e. the more the sum will already be of the form of a single Gaussian). This last paragraph is difficult to follow in the general case, and so

we illustrate the process for a single pixel in a cluster with two models overlapping in figure 3.9.

With this, we can look at the overlapping particles (1 and 2) from figure 3.7 in more detail. If target 1 and target 2 both compete for $S_t = x$, we could (as in the figure) consider the events incompatible. Alternatively, we could consider the possibility that both objects are in fact at x, using the above derivation to update each model. To this end, we construct a new cluster that is a union of the two single particle clusters with the appropriate prior, and update it accordingly. The three clusters ($\{P_1\}$,$\{P_3\}$,$\{P'_1,P'_3\}$,) are now all incompatible. The first considers the possibility that target 1 is at x, but target 2 is not (and vice versa for the second possibility), and the third considers the case that both are present. This is illustrated in figure 3.10.

With this established, we can summarise the inference process as shown in figure 3.11.

It is worth taking a moment to reflect on the slightly different process. In the standard BRE, the normalisation makes each particle compete with every other particle across the entire space. By changing our formulation to particles only competing against their relevant particles, we lose the fact that our PDF integrates to 1. At face value this feels as though we are losing some of the power of the normalisation. In reality, the opposite is true. The PDF integrating to 1 is a result of the exactly-1-target-present assumption. With our new formulation the PDF is unconstrained; each value is the likelihood that an object is at S_t with model M. If the background explains the observations well, this PDF will integrate to a small value. If there are multiple targets, it will integrate to more than 1. While it would be interesting to investigate this further on the synthetic problem for different numbers of targets with varying M values, we press on towards our core task.

3.4 Persistent Multiple Target Tracking

In the above formulation, the modelling aspect of SMAE is only applied to individual targets. The framework can be applied to much larger aspects of the adaptive tracking task. The variable M can hold information about the false positive rate, the camera movement model, the perspective model or any other imperfectly known property that affects the whole tracking task and which must be considered a unknown constant throughout the tracking task.

In our application of SMAE in the next chapter, we separate targets into objects of interest and clutter (this is addressed in detail in section 3.5.3). This comes down to including a term $p(I|M,Y)$: the probability that an object is of interest and not

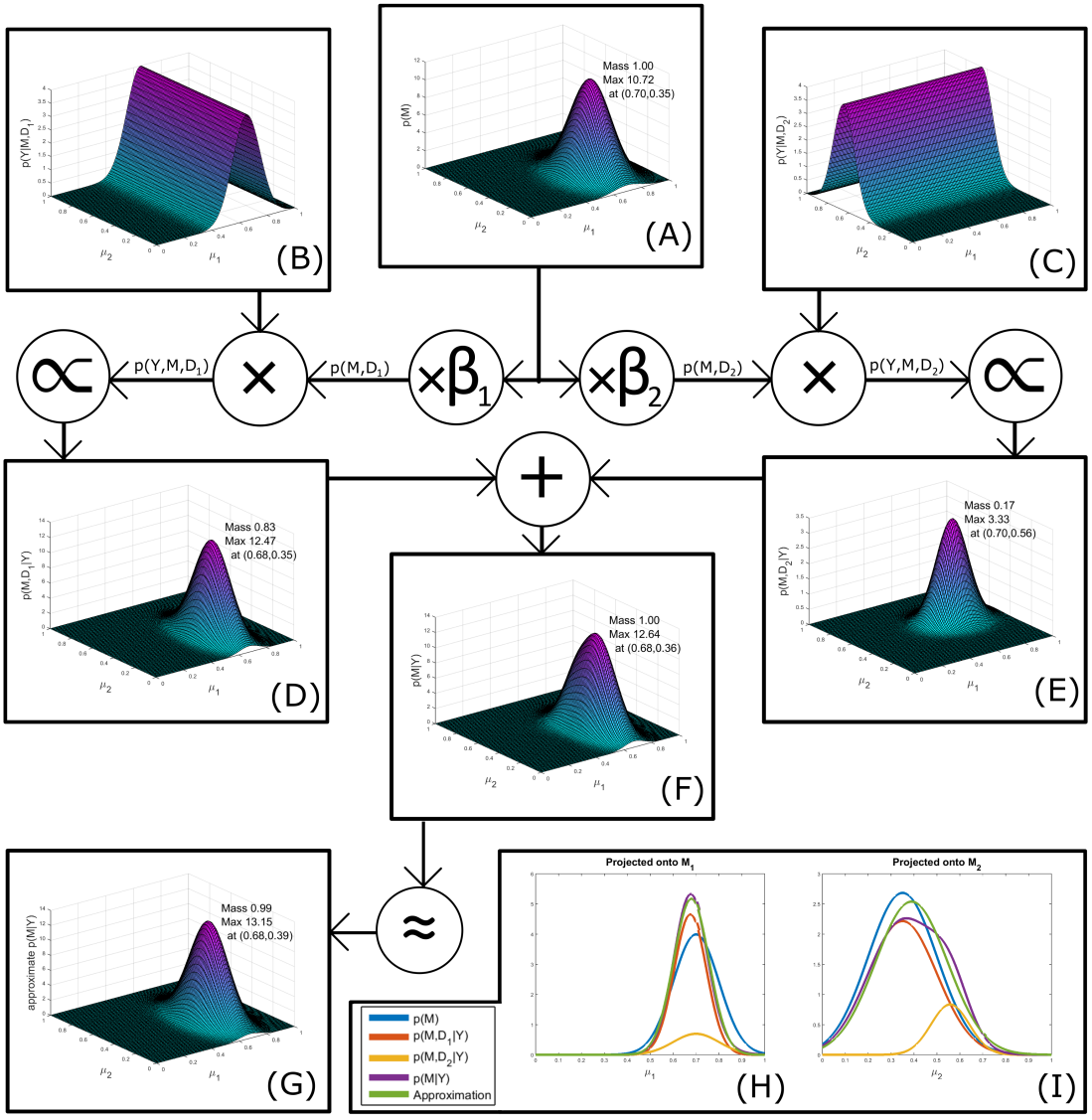
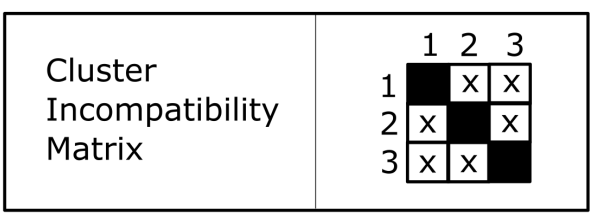


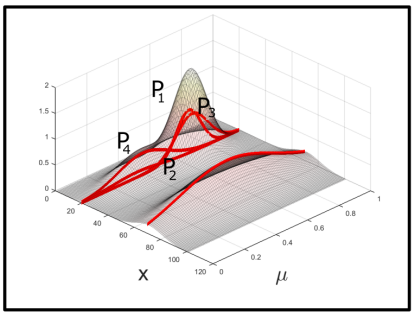
FIGURE 3.9: Model inference for overlapping particles. All surfaces shown are joint PDFs across the two particles' M-variables (their S-variables are the same, due to the overlap). We use the Cool color map (cyan to purple) to differentiate this from other joint surfaces previously shown, especially X the joint across a particle's (M, S_t) variables. The prior on the joint M space is shown in (A). This is constructed from independent Gaussian distributions for μ_1 and μ_2. The likelihoods (i.e. observation models) given that the observation originated from $\hat{X}_1$ and $\hat{X}_2$ are shown in (B) and (C) respectively. By taking the point-by-point product of these (with the appropriate β values), we can get the two terms in the numerator of equation 3.23. To get the normalising factor, we take the sum of the masses of the distributions in these two terms. Once we have divided by this normalisation constant, we get (D) and (E), the posteriors attached to D_1 and D_2 respectively, and their sum (F). We then approximate this posterior, with μ_1 and μ_2 being independent Gaussian variables, to get (G). On account of this final approximation, we can do this entire process without constructing the joint PDF. This process projected onto M_1 and M_2 is shown in (H) and (I). Key features to note are as follows: (B) and (C) only affect their relevant variable. The peak of (D) has moved in μ_1 and the Gaussian is narrower along μ_1's dimension, but the plot is unchanged in μ_2; equivalently for (E). The relative scales of (D) and (E) are due to how far the observation was from the priors on μ_1 and μ_2 respectively. This leads to a posterior (F), that is a sum of Gaussians. We accept the inaccuracy of a single Gaussian approximation (G), as (F) will become dominated by a single Gaussian for disparate M_1 and M_2 variables.

Particle	1	2	3	4
S_t	20	20	20	20
μ_{center}	0.68	0.56	0.68	0.39
Std	0.07	0.08	0.07	0.15
Mass	0.75	0.1	0.42	
Target ID	1	2	1	2
Cluster ID	1	2	3	3

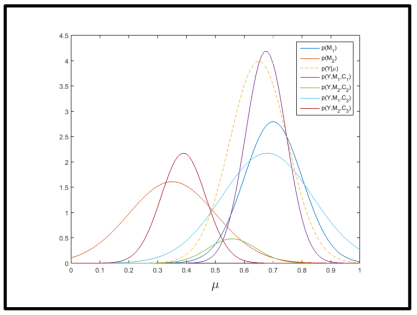
(A)



(B)



(C)



(D)

FIGURE 3.10: The illustration of cluster generation during particle overlap. Particle 1 and 2 represent particles 1 and 2 from figure 3.7; they are maintained to cover the event that only one target is at location x. A new cluster containing particle 3 and 4 (clones of particles 1 and 2 respectively) is generated to cover the event that both are present. In this example, the observed pixel value lies between the peaks of the two particles' distributions. This means that if only one particle is present then it needs to explain the pixel, and is updated more than the combined cluster, where each pixel may or may not have been responsible. In the combined case, target 1 (i.e. particle 3) is a much better fit, so it gets updated the most. The cluster's likelihood is calculated as a unit (hence only one value for the mass).

clutter, given the observations. Because our persistent tracker is gathering information constantly, we can include observations before initialisation. Thus at initialisation, our track's model has a prior that is a function of all previous observations and tracks. Even though this is still a distribution on M for a specific particle, all models prior to initialisation do not have specific observations (i.e. pixels selected from Y_t by S_t, to be explained by M). Thus a single pre-initialisation M_0 can be maintained, and each initialised track starts off with the current prior M_0.

This forms the difference between an adaptive tracker and a persistent tracker. An adaptive tracker exists only as long as its track is maintained, and all information gained in that track is discarded at the end. A persistent tracker's focus is on long-term performance, and so each track is part of the larger task. Much like an adaptive tracker uses each frame to keep its tracker optimal for the subsequent frames, a persistent tracker

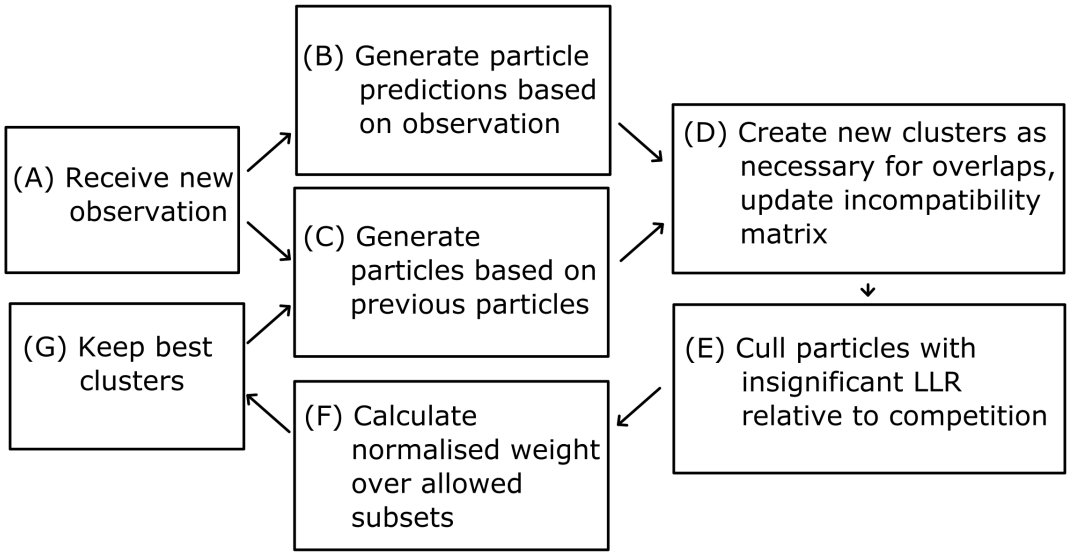


FIGURE 3.11: A single iteration of the particle-based SMAE tracking algorithm. For each frame that is received, (A) a set of hypothesised particles is created, using the pixels that are most salient (B) and the previous frame's output (C). The algorithm then updates the clusters, calculates their LLR, and creates new clusters as necessary (D). The algorithm culls those particles whose LLR are insignificant compared to those they are competing against (E). This means that when particles overlap, and clusters are created to handle the various possibilities, insignificant ones are disregarded. Finally, the algorithm uses equation 3.12 to calculate the posterior for each particle (F), and keeps the best ones for the next frame (G).

uses each track to ensure that the tracker is optimal for subsequent tracks.

3.5 Implementing the Framework for VOT

We have established the basic ideas underpinning our framework with a synthetic problem, to make the derivation more transparent. Now we transfer these ideas to a more standard template-based adaptive tracker. There are three things we need to build into the framework before we can use it on our maritime surveillance problem: exploring several observation models that may be appropriate for template based VOT; establishing a principled way to incorporate multi-pixel models that occur at multiple scales; and including the differentiation between objects of interest and clutter.

3.5.1 Different Observation Models

We consider three different sets of observation models[10]: a Gaussian distribution, a uniform distribution with an alpha mask, and a Gaussian distribution with an alpha

[10]We run them as separate test cases, but they could be run in parallel in a combined M-space.

mask.

We will establish our approximations for the 1-pixel synthetic problem mentioned above, before extending them to a more appropriate template system in section 3.5.2.

The straight Gaussian approach assumes an observation that is distributed around a mean μ with a standard deviation σ_{obs}. The parameter being estimated is μ, with σ_{obs} set at a constant value. This is the model used in the synthetic problem above. While it is possible to infer both μ and σ, we found no success with this in our maritime surveillance problem. The space required to present the resulting model's derivation would be unjustified for a negative result. We tested different σ_{obs} values, with many variations in the template layout, but found that they all suffered from a tendency to blur out the detail of a target (see section 5.1.3).

We present the next two models out of the order in which we tested them, for simplicity in derivation.

The observation model we finally chose is a uniform distribution with an alpha mask. The likelihood $p(Y_t^x|M)$ is parameterised by a probability α that the object will assert itself. If the model asserts itself, all pixel observation values are equally likely[11]; if it does not, the background model is used for the pixel observations. Thus the likelihood is

$$p(Y_t^x|M) = \alpha + (1-\alpha)p(Y_t^x|Bg_x), \tag{3.24}$$

with $p(Y_t^x|Bg_x) \sim \mathcal{N}(0, \sigma_{obs}^2)$. Figure 3.12 shows the update function using this likelihood. Because the function's scale changes radically for different values of Y_t^x, and the inference engine normalises the distribution across M, the update function shown is also normalised across M for clarity. For a given input Y_t^x at location x, each M is updated by multiplication with the update function at (Y_t^x, M). The key observation is that the update function transitions quickly from $update = 1-\alpha$ to $update = \alpha$.

Now we need to find a prior, such that multiplication with the update does not change its form as a posterior (which will of course be the prior for the next step). If we assume all previous update steps were dominated by either the foreground observation model (i.e. they were sufficiently far from 0), or that they were dominated by the background observation (i.e. they were sufficiently close to zero), then our prior will be of the form

$$p(M) \propto \alpha^{nPos}(1-\alpha)^{nNeg}, \tag{3.25}$$

where *nPos* is the number of historical samples dominated by the foreground model, and *nNeg* is the number dominated by the background. This will have a maximum

[11] Pixel values are scaled to range from 0 to 1, making the PDF equal to 1.

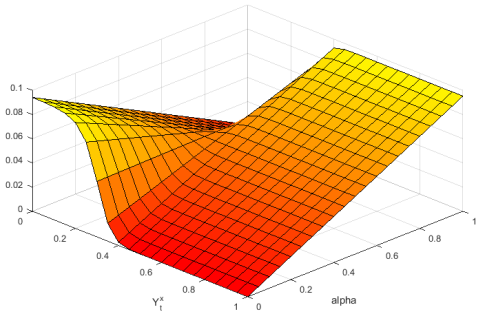


FIGURE 3.12: Update function for a uniform observation model with an alpha mask (normalised across α). For an observed pixel Y_t^x, the distribution on α will be multiplied by the function defined by the plane for Y_t^x. The observation standard deviation σ_{obs} is set to 0.2 for this image.

at $\alpha = \frac{nPos}{nPos+nNeg}$. We will consider the three main cases upon a new pixel Y_t^x being observed. Firstly, the foreground model could dominate and the update function can be approximated by $p(Y_t^x|M) = \alpha$. This would add another case to $nPos$, putting the new maximum at $\frac{nPos+1}{nPos+nNeg+1}$. Secondly, the background model could dominate and the update function could be approximated[12] by $p(Y_t^x|M) = 1 - \alpha$. This would add a case to $nNeg$, putting the new maximum at $\frac{nPos}{nPos+nNeg+1}$. Finally, if neither dominates, the function could be approximated as $p(Y_t^x|M) = 1$. This would not update the prior at all, so the new maximum would still be at $\frac{nPos}{nPos+nNeg}$.

It would be convenient for us if this last case put the maximum at $\frac{nPos+0.5}{nPos+nNeg+1}$, because then we could define

$$(nPos_{new}, nNeg_{new}) = (nPos, nNeg) + (p(Y_t^x|Fg_p), p(Y_t^x|Bg_p))\frac{1}{p(Y_t^x|Fg_p) + p(Y_t^x|Bg_p)}. \tag{3.26}$$

This would transition smoothly between our three cases. The difference between the accurate approximation $\frac{nPos}{nPos+nNeg}$ and our desired approximation $\frac{nPos+0.5}{nPos+nNeg+1}$ is emphasised when $nPos + nNeg$ is not large (i.e. when few observations have been made). In this case, our approximation pushes the mode of α's distribution towards 0.5, and requires more evidence in the future to push the mode to either limit. In this way, inconclusive evidence acts as evidence against both limits, especially when little historical information is available. This is not a problematic error to have: in cases where historical information is lacking and the current evidence is inconclusive, the system does not commit to either extreme and requires more evidence to commit in the future[13]. With this,

[12] These update functions are normalised across M (that is α), so leaving out constant multiples does not affect the result.

[13] This is exactly the sort of ad hoc reasoning that Jaynes [2] rails against, and which we are trying to avoid. However, as discussed in section 1.1, while our goal is to work as close to the principled side of the spectrum as possible, we will be forced to take steps toward the practical side in order to have

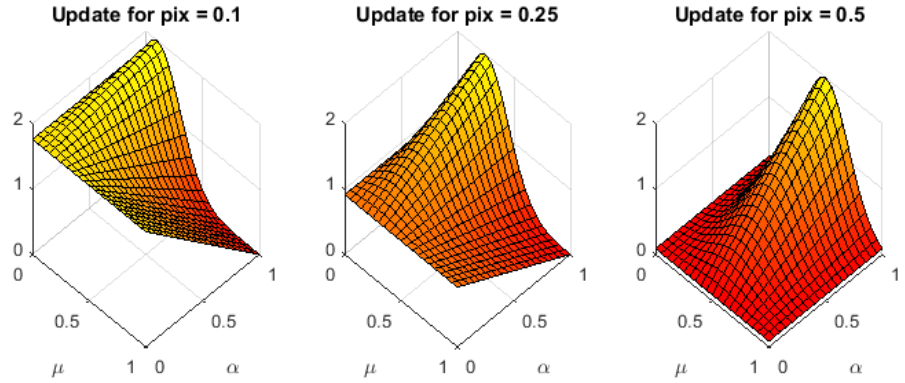


FIGURE 3.13: Update function for a Gaussian observation with an alpha mask. The standard deviation on observations (σ_{obs}) is set to 0.2 for this image. For low values of Y_t^x, the graph slopes up from the μ-z plane when μ is near Y_t^x and downwards when μ is far from Y_t^x. For larger values of Y_t^x, the value at the μ-z plane decreases. This can be interpreted as follows. If Y_t^x is near where μ is believed to be, a foreground pixel has probably been observed; higher values of α are more likely, and the μ should be closer to Y_t^x. If Y_t^x is far from where μ is believed to be, a background pixel has probably been observed (as long as background pixels are still feasible) and lower values of α are more likely. If background pixels are not feasible, then μ should be updated heavily towards the observed Y_t^x.

we can now summarise our distribution $p(M|Y_{1:t})$ with the ordered pair $(nPos_t, nNeg_t)$, where

$$(nPos_{t+1}, nNeg_{t+1}) = \left(nPos + \frac{1}{1 + p(Y_{t+1}|Bg)}, nNeg + \frac{p(Y_{t+1}|Bg)}{1 + p(Y_{t+1}|Bg)}\right), \tag{3.27}$$

and the mode of α at any time t can be found at $\frac{nPos_t}{nPos_t + nNeg_t}$.

This is the observation model with which we had the most success, but in the process of finding it we first tried an observation model that combined the first two described observation models, namely a Gaussian observation model with an alpha mask:

$$p(p_i|M) = \alpha p(Y_t^x|M) + (1 - \alpha)p(Y_t^x|Bg), \tag{3.28}$$

with $p(Y_t^x|M) \sim \mathcal{N}(\mu, \sigma_{obs})$ and $p(Y_t^x|Bg) \sim \mathcal{N}(0, \sigma_{obs})$. Here both μ and α are inferred. This means that our update function has an output in the two-dimensional space (μ, α), so we cannot show the observed pixel on the same axes. Instead, figure 3.13 shows some update functions for sample pixel values.

If we combine the last two approaches, we can represent μ in a Gaussian distribution, and α as an ordered pair $(nPos, nNeg)$ as though the two variables were separable. At each update we take local approximations of μ and α, and combine the result into the

tractable solutions. In this case, we believe the gain is worth the cost, as the error is most apparent in the cases where little evidence has been gathered, and the direction of that error is favourable.

best approximation of the aforementioned form. This will cause problems if the model drifts, as implicit in the prior for any step is the local approximations of all former steps. In practice we found this to be a significant problem, but we found no other feasible representations for this PDF. We will not go into the derivation of this in detail, as ultimately we found the model ineffective, but the final result was to combine the two approximations. We calculate the Gaussian approximation of μ's update (for the foreground case) and prior (for the background case), using their scale for updating $(nPos, nNeg)$, and update μ in a way similar to equation 3.23. The algorithm we derived is:

- $nPos_{Fg} = nPos_{prior} + 1$
- $nNeg_{Fg} = nNeg_{prior}$
- $\sigma_{Fg} = \left(\sigma_{prior}^{-2} + \sigma_{obs}^{-}2\right)^{-\frac{1}{2}}$
- $\bar{\mu}_{Fg} = \sigma_{Fg}^{2}\left(\sigma_{prior}^{-2}\bar{\mu}_{prior} + (\sigma_{obs}^{-2}.pix\right)$
- $S_{Fg} = \frac{nPos+1}{nPos+1+nNeg} e^{-0.5\left((\frac{\bar{\mu}_{prior}}{\sigma_{prior}})^2 + (\frac{pix}{\sigma_{obs}})^2 - (\frac{\bar{\mu}_{Fg}}{\sigma_{Fg}})^2\right)} mass_{prior}$
- $nPos_{Bg} = nPos_{prior}$
- $nNeg_{Bg} = nNeg_{prior} + 1$
- $\sigma_{Bg} = \sigma_{prior}$
- $\bar{\mu}_{Bg} = \bar{\mu}_{prior}$
- $S_{Bg} = \frac{nPos}{nPos+1+nNeg} e^{-0.5(\frac{pix}{\sigma_{obs}})^2} mass_{prior}$
- $(nPos_{posterior}, nNeg_{posterior}) = (nPos_{prior} + \frac{S_{Fg}}{S_{Fg}+S_{Bg}}, nNeg_{prior} + \frac{S_{Bg}}{S_{Fg}+S_{Bg}})$
- $(\bar{\mu}_{posterior}, \sigma_{posterior}) = \left(\frac{S_{Fg}\bar{\mu}_{Fg}+S_{Bg}\bar{\mu}_{Bg}}{S_{Fg}+S_{Bg}}, \frac{S_{Fg}\sigma_{Fg}+S_{Bg}\sigma_{Fg}+(\bar{\mu}_{Fg}-\mu_{posterior})^2+(\bar{\mu}_{Bg}-\bar{\mu}_{posterior})^2}{S_{Fg}+S_{Bg}}\right)$
- $mass_{posterior} = S_{Fg} + S_{Bg}.$

There is very little value in including these rules with such a brief explanation, however their negative results do not justify further space. We include this observation model, as it is an obvious choice in the light of the other two options, and the negative results are informative, It can be derived from the description above.

3.5.2 Getting Templates to Work

There are two things we need to incorporate to make the current framework applicable to template-based VOT: incorporating multiple views, and managing a multi-pixel view that can occur at multiple scales.

Incorporating multiple views involves managing multiple templates per particle. The observation model is then split between the views. While it would be possible to handle the inference as an approximation of the sum of distributions for each of the templates being activated, this would lead to a nasty combinatorial problem when multiple templates overlap, each with multiple views. As this is in the innermost loop of our algorithm, we choose simply to use the view which provides the biggest contribution (and is therefore the best fit). In this way, we only update the most relevant view for each particle.

Switching from pixels to templates is a slightly more involved problem. For the Gaussian observation models, each view in the observation model of M contains the parameters $\vec{\mu}$, a n^2-element vector containing the means for the n by n grid of pixels. Let Y be the observation in the r by c window cropped by S_t, which we write as a rc element vector $\vec{y}$. In order to address the discrepancy of scale between M and Y, we consider a super-resolution image of the scene $\vec{u}$ at resolution nr by nc. The lower-resolution pixels can be expressed as averages of the super-resolution pixels associated with them. We write this linear transformation as $\vec{y} = W\vec{u}$ and $\hat{\vec{y}} = V\vec{u}$, where $\hat{\vec{y}}$ is the image rescaled to M's scale, and both V and W are wide matrices[14]. Neither V nor W is invertible, but as we are trying to make the best inference possible we proceed with the pseudo-inverse. The end result will make sense in terms of a general linear transformation taking $\vec{y}$ to $\hat{\vec{y}}$. For V and W describing averaging of sub-pixels, we get $V^{\#} = rcV^T$ and $W^{\#} = n^2W^T$. With this we can make a projection for $\vec{\mu}$ into Y's space as $\hat{\vec{\mu}} = WV^{\#}\vec{\mu}$. Here $\hat{\vec{\mu}}$ is the expected mean of the Gaussian observation distribution at the locations of the elements of $\vec{y}$.

At this point there is a choice to be made. We have decided to use a set standard deviation for σ_{obs}[15]. Should we set σ_{obs} to be constant on the scale of M or of Y? At the end of this section we make an argument for the latter, thus we continue with this choice. We are considering

$$p(M|Y) = \frac{p(Y|M)p(M)}{p(Y)}. \tag{3.29}$$

[14] In other words, they have more columns than rows.

[15] Remember that this is the σ for the difference between a given pixel observation and its underlying value, not the standard deviation of our certainty on that underlying value.

Our observation model can be written as the multivariate normal distribution measuring the distance from each observed pixel to its respective predicted $\hat{\vec{\mu}}$:

$$p(Y|M) = \frac{1}{\sqrt{2\pi|\Sigma|}^{rc}} e^{-0.5\left((\vec{y}-\hat{\vec{\mu}})^T\Sigma^{-1}(\vec{y}-\hat{\vec{\mu}})\right)}. \tag{3.30}$$

Because we have all pixels independent and with the same standard deviation, Σ will be of the form $\sigma_{obs}^2 I$. Along with our prediction for $\hat{\vec{\mu}}$, we get

$$p(\vec{y}|M) = \frac{1}{\sqrt{2\pi\sigma_{obs}^{2rc}}^{rc}} e^{-0.5\left((\vec{y}-WV^{\#}\vec{\mu})^T(\sigma_{obs}^{-2}I)(\vec{y}-WV^{\#}\vec{\mu})\right)}. \tag{3.31}$$

Assuming pseudo-inverses give the best possible inverses for our inference,

$$p(\vec{y}|M) = \frac{1}{\sqrt{2\pi}^{rc}\sigma_{obs}^2} e^{-0.5((VW^{\#}\vec{y}-\vec{\mu})^T(V^{\#})^TW^T(\sigma_{obs}^{-2}I)WV^{\#}(VW^{\#}\vec{y}-\vec{\mu})}. \tag{3.32}$$

However, considering that $W^T = \frac{W^{\#}}{n^2}$ and $(V^{\#})^T = rcV$, and that multiplication with constants and the identity matrix is commutative,

$$p(\vec{y}|M) = \frac{1}{\sqrt{2\pi}^{rc}\sigma_{obs}^2} e^{-0.5\left((VW^{\#}\vec{y}-\vec{\mu})^T(\sigma_{obs}^{-2}I)\frac{rc}{n^2}(VW^{\#}\vec{y}-\vec{\mu})\right)}. \tag{3.33}$$

This finally yields

$$p(\vec{y}|M) = \frac{1}{\sqrt{2\pi}^{rc}\sigma_{obs}^2} e^{-0.5\left((VW^{\#}\vec{y}-\vec{\mu})^T\left(\left(\sigma_{obs}\cdot\frac{n}{\sqrt{r.c}}\right)^{-2}I\right)(VW^{\#}\vec{y}-\vec{\mu})\right)}. \tag{3.34}$$

This is a Gaussian in $\vec{\mu}$ with independent standard deviations of $\sigma_{obs}\frac{n}{\sqrt{rc}}$. The normalisation constant does not matter, as it is constant for a given $\vec{y}$ and will be normalised out in the inference. We call the projection $VW^{\#}\vec{y}$ of the observations into M's scale $\hat{\vec{y}}$. If we set our prior on M to be a Gaussian (with mean $\bar{\vec{\mu}}_{prior}$ and variance Σ_{prior}), it will make the posterior at each step a Gaussian. Looking again at the inference

$$p(M|Y) = \frac{p(Y|M)p(M)}{p(Y)}, \tag{3.35}$$

and using the standard result for multiplying normal distributions, this becomes a Gaussian distribution with mean

$$\bar{\vec{\mu}}_{posterior} = \Sigma_{posterior}\left((\Sigma_{prior}^{-1})\bar{\vec{\mu}}_{prior} + (\sigma_{obs}^2\frac{n^2}{rc})^{-1}I\hat{\vec{y}}\right) \tag{3.36}$$

and variance

$$\Sigma_{posterior} = (\Sigma_{prior}^{-1} + (\sigma_{obs}^2 \frac{n^2}{rc})^{-1} I)^{-1}. \tag{3.37}$$

The end result of the above derivation is that the model inference can be done with a rescaled version of the observation patch, scaling σ_{obs} by $\frac{n^2}{rc}$.

A different method is needed for the observation models that use alpha masks. The full observation model is of the form

$$p(Y|M) = \prod_{Y \text{ resolution}} (\alpha_{pred} p(y|M) + (1 - \alpha_{pred}) p(y|Bg)) \tag{3.38}$$

for some α_{pred} predicted from our model. For our inference we want it to be of the form

$$p(Y|M) = f\left(\prod_{M \text{ resolution}} \alpha p(y_{rescaled}|M) + (1 - \alpha) p(y_{rescaled}|Bg) \right) \tag{3.39}$$

for some function f and rescaled versions of the input image. If we consider underlying continuous surfaces α_{cont} and y_{cont}, and the function

$$g(pix) = \alpha_{cont}(pix) p(y_{cont}(pix)|M) + (1 - \alpha_{cont}(pix)) p(y_{cont}(pix)|Bg), \tag{3.40}$$

we get the following two approximations for the geometric mean of g:

$$GeometricMean \approx \sqrt[rc]{\prod_{Y \text{ resolution}} (\alpha_{pred} p(y|M) + (1 - \alpha_{pred}) p(y|Bg))} \tag{3.41}$$

and

$$GeometricMean \approx \sqrt[n^2]{\prod_{M \text{ resolution}} (\alpha p(y_{rescaled}|M) + (1 - \alpha) p(y_{rescaled}|Bg))}. \tag{3.42}$$

Putting these together, we get

$$p(Y|M) \approx \prod_{M \text{ resolution}} (\alpha + (1 - \alpha) p(y_{pred}|Bg))^{\frac{rc}{n^2}}. \tag{3.43}$$

Given that integral values of $\frac{rc}{n^2}$ amount to repeated updates, it makes sense to let the weight added to *nPos* or *nNeg* be scaled by $\frac{rc}{n^2}$. This also has an intuitive interpretation: given *nObs* observation pixels to be spread over *nModel* model pixels, each one receives $\frac{nObs}{nModel}$ amount of information. That information is spread between foreground (*nPos*)

and background (*nNeg*) according to the observation models, and the final approximation of α is $\frac{nPos}{nPos+nNeg}$.

For the observation model that combines an alpha mask with a Gaussian observation distribution, we use a combination of the above approximations. The final result is to resize the observation Y to n-by-n pixels, using the modified $\sigma_{\text{effective}} = \sigma_{obs}\frac{n}{\sqrt{rc}}$ for the Gaussian observation (which is then combined with the prior according to the relative performance of the foreground and background models), scaling the additions to *nPos* and *nNeg* by $\frac{rc}{n^2}$.

The last topic to address is the discussion about the scale for σ_{obs}. In the two Gaussian models above, we chose σ_{obs} to be constant at the observation scale rather than at the model scale. We will now justify that decision.

We showed in our discussion of Gaussian observation models that a pixel observation standard deviation at Y's scale of σ will correspond to one in at M's scale of $\sigma\frac{n}{\sqrt{rc}}$. For want of a concrete example, let us consider a case in which there are four times as many pixels in the observation as there are in the model. A pixel standard deviation of 0.1 at M's scale would correspond to a pixel standard deviation of 0.2 at Y's scale. This makes sense. We have 4 $\vec{y}$ pixels per $\vec{\mu}$ pixel and they are added together; when we add random variables, we get a regression to the mean.

We can argue this both ways. Assuming firstly that discrepancy from the mean during the observation is a function of the target's surface due to changes such as lighting and angle, it would make sense to use a scale that is constant for a target, not changing as the object moves closer to the horizon. Thus more target surface being aggregated into each observation pixel would lead to a regression to the mean, and hence a smaller σ_{obs}. This would suggest a constant observation sigma on M's scale.

On the other hand, assuming that the discrepancy from the mean during the observation is a function of the photographic process, it would make sense that every pixel deviates from its true value due to a normally distributed error function. This suggests a constant observation sigma on Y's scale.

Ultimately, we chose to set the standard deviation on the observations constant on Y's scale as we believe that our 'unit' of information should be in pixels. When a particular observation of a target has relatively few pixels, the underlying inference on M will use a broader Gaussian, updating the model less. As we expect, distant views of an target will update our knowledge less than closer views.

3.5.3 Separating Interesting Objects and Clutter

The last addition we make to the framework is to differentiate between objects of interest and clutter. While it would be possible to classify those objects that are not of interest as background, and expect the observation model to reject them, we believe it is more useful to acknowledge clutter as trackable targets and then train our tracker to favour targets that are more likely to be of interest.

In the next chapters we apply SMAE to the problem of maritime surveillance. The sea is a noisy background, and many of the artefacts that make up the background (static rocks, wake, waves, birds) have more in common with the targets we will track (they are patches of localised salience relative to surroundings that survive over a number of frames) than the general background (which tends to be easily rejected by a simple salience map). Thus, it is easier to separate the background from targets, and then separate targets of interest from clutter, than it is to separate targets of interest from both clutter and background (this is much like Teutsch and Krüger [41], who use two classifiers).

To accommodate this into our framework, we include the variable I to represent the event that the target in question is an object of interest:

$$p(I, M, S_t|Y_{1:t}) = p(I|M, S_t, Y_{1:t})p(M, S_t|Y_{1:t}). \tag{3.44}$$

Here the probability that an object is of interest given that it exists $p(I|M, S_t, Y_{1:t})$ is only a function of our knowledge of M at time t[16]. This makes

$$p(I, M, S_t|Y_{1:t}) = p(I|M, Y_{1:t})p(M, S_t|Y_{1:t}). \tag{3.45}$$

Here the second term is the posterior we have been using until this point. Thus we can track our particles as described above, and use an interest model in M that is updated in a principled manner over all observations to separate clutter from targets of interest.

We mentioned in section 3.4 that we can include in M all the information we have learned in previous tracks prior to initialisation. We change the subscript on Y to start at $-\infty$ to show that we are learning from previous tracks to improve our understanding of which M are more likely to represent targets of interest. Thus we move from an adaptive tracker (focusing on a single track) to a persistent tracker (focusing on improving over

[16] I.e. we are dropping S_t in the next equation. This is because the "interest" in a given particle is based on it's appearance not its location

consecutive tracks):

$$p(I, M, S_t|Y_{-\infty:t}) = p(I|M, Y_{-\infty:t})p(M, S_t|Y_{-\infty:t}), \tag{3.46}$$

where $p(I|M, Y_{-\infty:t})$ uses all the information available to the tracker over its deployment lifetime to model the probability that a particular model is a target of interest.

3.5.4 Conclusion

In this chapter we used a simple synthetic problem to guide us in our derivation of a Bayesian framework that both estimates state and model parameters (SMAE). We moved from single target to multiple targets, incorporated the persistent tracking task in the framework, and finally extended the framework to be usable on VOT tasks.

With the derivation complete, we move to a challenging task on which to apply SMAE.

Chapter 4

Maritime Surveillance Data Set and Salience Filter Design

Over the next two chapters we apply SMAE to a concrete problem: maritime surveillance. Our goal here is to show that SMAE creates tractable competitive solutions. The core contributions in our work are in the previous chapter. However, to present just a derivation and the philosophical discussion in chapter 7 would give the impression that nothing useful has been produced. For this reason, we present an application of SMAE with tangible results, to verify our contribution.

We split this application across two chapters, as the ground we must cover before reaching the adaptive and persistent tracking is not directly dependent on our framework. We start this chapter in section 4.1, with a discussion of the task at hand, and a definition of the problem. We discuss the data set we use to test our framework in section 4.2. It would be possible to include the salience filter into the SMAE formulation, learning the relevant parameters while completing the rest of the task. While this would be preferable, the tracking task is already complex and our purpose is only to demonstrate an application of SMAE. For this reason we implement our salience filter as a pre-filter, presenting its output to the SMAE tracker as Y_t. We present our work on the salience filter in section 4.3 and address the rest of the tracking task in chapter 5.

4.1 Problem Definition

Moreira et al. [21] define maritime surveillance as 'the effective recognition of all maritime activities that impact the security, the economy or the environment'. Harbours and

cage aquaculture sites need information about vessels in the immediate and surrounding areas to protect from accidental and malicious harm. Traditional CCTV systems rely on human operators not becoming weary or distracted, making automated vessel detection desirable. Several active systems exist (ARGOS [10], MAAW [22], and DeMarine-DEKO [23], yet it is still an ongoing research task.

It is worth noting that we are not choosing a difficult visual problem that radar could easily solve. While radar works well on large metal vessels, in maritime surveillance we cannot ignore small inflatable or wooden vessels. These are often easier to detect in the visual spectrum. Krüger and Orlov [50] use the infra-red spectrum to detect vessels. Our approach would synergise well with this use of IR, as our framework could just as easily work on an infra-red feed as on a visible spectrum feed, and Bayesian frameworks lend themselves towards data fusion.

What makes maritime surveillance a good application for our SMAE framework is the noise that fills the background. While humans have developed rich priors that help us detect anomalies on bodies of water in a video sequence, it is difficult to isolate all the visual cues that we use to rule out waves as potential objects. The wave noise creates a perfect opportunity for model learning to improve rejection of false positives, simultaneous to the tracking task.

Because the waves form the basis of the challenge, the choice of data heavily influences the difficulty of the tracking problem. We note that many systems are tested on sequences with flat waters that are largely homogeneous in the visual field, and that the boat size is often an appreciable fraction of the frame. We show the strength of our approach by testing it on more challenging data. We draw further attention to this in section 4.2.

We also draw attention to the difference in mindset between designing an adaptive tracker and designing a persistent tracker. In designing an adaptive tracker, our focus is on creating a system that completes the tracking task. That is, once it has been initialised on a single target, it must not drift. For a persistent tracker, we want a system that solves the long-term problem. It must auto-initialise, be able to track multiple targets, and improve its performance over time. When focusing on adaptive tracking, the learning task is the appearance of the current object. In persistent tracking, we focus on learning the appropriate priors to take into each track.

In light of maritime surveillance being both a useful and a difficult visual tracking task, we select it as a good sample application of SMAE. This forms the two goals that guide our design decisions: firstly, to make an effective persistent tracker for maritime

surveillance; secondly, in doing so, to test whether SMAE is viable on challenging real-world problems. With these two goals in mind, we make some constraints on our input. We constrain ourselves to sequences with no camera movement. This enables us to have annotated horizon lines, and is a realistic constraint for surveillance. It would be possible to include camera motion as a variable inside M, however our scope is large enough without including this. Secondly, we focus on tracking boats against the ocean, rather than against the ocean, sky and distant shore. This is justified for surveillance, as we are most concerned about small vessels that radar would miss, and placing the camera high enough is always an option and usually a good idea. In light of this choice, we will only need to build salience filters for the ocean. This is a detailed process, and could be completed for the sky and for static background. However, the ocean is the more challenging background, and will serve as a proof of functionality without us devoting more time and space repeating very similar processes.

Finally, we present our task in a concise manner: given an extended video sequence of a maritime scene from a static camera, detect and track objects that are not ocean, and improve tracking performance over time.

4.2 Data Set

In this section we address the data sequences on which our persistent tracker will be operating. We start by addressing some of the problems related to choosing a data set, then we present the data set we will use, and finally we discuss our choice of data and how it will be used.

4.2.1 Difficulties in Choosing a Data Set

The choice of data for the data set has a far-reaching effect. All algorithms will have strengths and weaknesses, and the relative frequency between different events in the data can skew the results. This is exacerbated by the fact that it is impossible to say what an unbiased result would be. We are trying to solve a general problem, presenting a solution that should work for any foreign object on any maritime scene. Unfortunately, to rank performances on this general problem, we must decide on the severity of different faults that occur in different instantiations of the general problem. To do this we must define the different priors with which to consider both fault conditions and problem instantiations. Our decision is implicit in our choice of data set. Is dealing with glare off the water important? Is tracking smaller boats more important than tracking larger

boats? With what regularity does inclement weather occur? The events that occur more often in our data will receive higher weight in our performance evaluation.

We can use our application as a guiding principle. We are designing a surveillance system, so our tracker should be sensitive toward (and thus our data set be loaded with) events that should be reported by a surveillance system. The most important situations to correctly identify are the most difficult and rare ones, such as covert malicious activities. This implies that our data set should have a high density of difficult situations. However, this would favour trackers that have a strong prior towards edge cases.

To see the danger in this line of thinking, consider a system tracking an egg in a magician's hand. As the sequence proceeds the magician passes the egg to his other hand, switching it with a decoy and palming the original. The more complex a tracker is, and the more prior information it has, the more problems it will have as the sequence progresses. Perhaps the tracker has enough of a prior that it recognises an object being passed from hand to hand, and continues to track the decoy. Perhaps the tracker recognises the momentary disappearance of the egg behind the hand as a potential occlusion, and maintains a possibility that the egg remains occluded, tracking the hand. Perhaps the tracker recognises the magician's attire to be indicative of an entity that causes edge cases. Perhaps the magician knows the audience will be suspicious and so opens with many concealing actions, forcing the tracker's PDF to be spread thinly over conflicting hypotheses before the real trick happens. What would we consider the best tracker behaviour in this case?

Focusing on the edge cases sets up a tracker to be tricked by more edge cases. In setting up a persistent tracker, we may want to enable it to learn to ignore common background behaviour. However, this makes it vulnerable to someone surreptitiously 'training' it to ignore behaviour that will be used maliciously in the future.

This is all to say that choosing data for training and testing a system is a non-trivial activity, and reporting the performance of a tracker on any data without a clear explanation of the data is dangerous although it is common in maritime surveillance literature[1].

We consider the following events relevant for a maritime surveillance data set:

- A variety of boat sizes,
- Boats which cross paths,
- Birds passing through the scene,

[1]This is regrettable, but understandable due to space limitations in publications, and to IP considerations with some data sets.

- Boats in static positions,
- Items which are not of interest in the water (e.g. lighthouses, reefs, etc.),
- Different lighting conditions including glare, non-uniform lighting across the ocean, and changes due to cloud cover,
- Target out-of-plane rotations, and
- Different ocean conditions.

We will make one more observation before describing the data set we selected. It is accepted that a system should not be tested with the same data on which it was trained. This is especially relevant in our context, in which the priors attached to these different events in the wild are unknown, yet implicit in the data. However, as with any system, ours was developed through iterations. The system will thus have been unintentionally optimised towards the data, despite our best efforts. This is not unique to us, and is implicit in any publication, however it is worth noting that often the rules we set in place to get meaningful results in machine learning are impossible to follow fully.

4.2.2 Description of Data Set

Our data set was generated at a number of South African ports by the CSIR's Optronic Sensor Systems group as part of the PRISM project, and is available from http://prism.csir.co.za/.

One of the assumptions we make in our tracker is a static camera (justified above). Because of this we segmented the original data, which contained some camera movement, into 22 different sequences where the camera is static for each sequence. A sample frame from each sequence is shown in figure 4.1. Table 4.2 presents a summary of the challenges in each data sequence. For each sequence a group is shown, to indicate which videos were segmented from the same sequence in the original data set. Each sequence has been annotated with the horizon line, a pixel mask of the ocean, and has had all salient objects manually labelled with a bounding-box and annotated as {boat, bird, stationary object, rocks, other}. The bounding-boxes are generous; we are testing self-initialising trackers, and different trackers include different borders around the object. In the maritime context knowing that there is a foreign object, and where it is, is more relevant than having a tight bounding-box.

(A) sequence 001

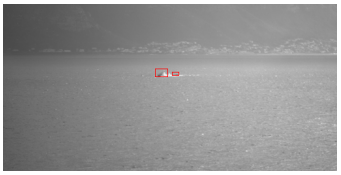
(B) sequence 002

(C) sequence 003

(D) sequence 004

(E) sequence 005

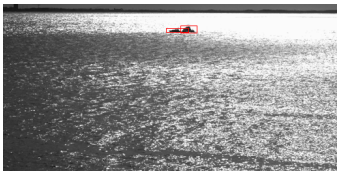
(F) sequence 006



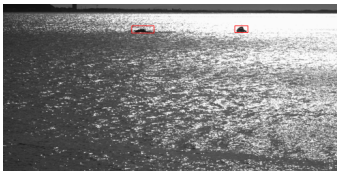
(G) sequence 007

(H) sequence 008

(I) sequence 009



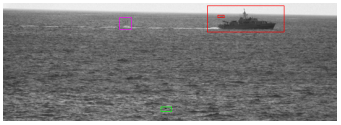
(J) sequence 010



(K) sequence 011

(L) sequence 012

(M) sequence 013



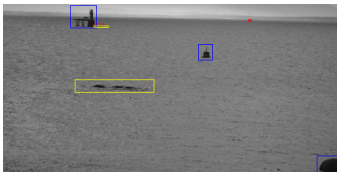
(N) sequence 014

(O) sequence 015

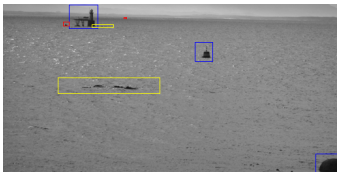
(P) sequence 016

(Q) sequence 017

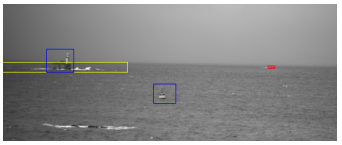
(R) sequence 018



(S) sequence 019



(T) sequence 020



(U) sequence 021

(V) sequence 022

FIGURE 4.1: Sample frames from each sequence in the data set. Red rectangles mark boats; blue rectangles mark stationary objects; green rectangles mark birds; yellow rectangles mark rocks and white water splashing on them; and magenta rectangles mark other objects that are difficult to distinguish (unlikely for a tracker to track).

Sequence	Group	#Targets	#Frames	Boat Size	Waves	Ocean lighting	Occlusions	Clutter
1	1	3	784	Small	Average	Non-uniform	No	Bird
2	1	2	977	Small	Average	Non-uniform	No	None
3	2	2	120	Small	Average	Uniform	No	Flock of small birds
4	2	3	441	Small	Average	Uniform	No	Bird, Flock of small birds
5	2	3	379	Small	Average	Uniform	No	Bird, Flock of small birds
6	3	2	642	Small	Low Contrast	Non-uniform	No	None
7	3	2	484	Small	Low Contrast	Non-uniform	Yes	None
8	3	2	584	Small	Low Contrast	Non-uniform	No	None
9	4	3	833	Both	Average	Non-uniform	Yes	None
10	5	2	142	Small	High contrast	Glare	Yes	Static Boat
11	5	2	304	Small	High contrast	Glare	No	Static Boat
12	5	1	216	Small	High contrast	Glare	No	None
13	5	2	196	Small	Average	Glare	No	Bird
14	6	4	490	Both	Average	Uniform	Yes	Bird
15	6	4	509	Both	Average	Uniform	No	Bird
16	7	3	435	Small	Average	Uniform	Yes	None
17	8	3	1798	Both	Average	Uniform	No	Buoy
18	9	3	1544	Small	Average	Non-uniform	Yes	None
19	10	7	1855	Small	Average	Non-uniform	Yes	Rocks, Buoy, Lighthouse
20	11	10	1547	Small	Low Contrast	Uniform	Yes	Rocks, Buoy, Lighthouse, Bird
21	12	8	1038	Small	Low Contrast	Non-uniform	No	Rocks, Buoy, Lighthouse, Bird
22	13	3	555	Small	Average	Non-uniform	No	Buoy

TABLE 4.2: Summary of data sequences.

4.2.3 Comments on Data

Our ultimate goal is to design a persistent tracker. To this end, it would have been preferable had our data been in the form of a few very long sequences, rather than many shorter sequences. However, the difficulties involved with imperfect data are not uncommon.

We wish to simulate persistent tracking. This means giving the tracker the opportunity to learn and improve on a time scale larger than that of an adaptive tracker. There is a fine balancing act between giving the tracker enough information to learn the appropriate models, and testing on the training data. We have data sequences from the same location. On one hand, these present similar information. Should they be considered too similar, thus equivalent to training and testing on the same data? On the other hand, this is exactly the sort of deployment we have in mind throughout our design: a tracker in a fixed location seeing only a particular scene. We are aware of how unreliable results achieved through testing on training data are, and so err on the side of caution. Sequences that were cropped from the same original video are not used to train the persistent tracker for each other. This simulates having watched similar situations without having direct information on a given sequence.

Another problem is the initialisation of the salience filter. Many salience filters need a baseline history. In a persistent tracker, this baseline is always available from the previous tracking. In our case we have many short sequences. In order to address this, whenever initialising salience filters we use a selection of frames spread over the sequence in question. While this does break the 'testing on training data' rule, initialisation is usually to train background subtraction. Any artefacts caused by future foreground objects on the initialisation have equivalent artefacts caused by past foreground objects. We will discuss initialisation for the salience filters in detail for each filter; ultimately the difference between priming a salience filter with average future information and average past information was not identified as relevant for our sequences.

We note that our sequences favour small boats (i.e. boats which affect relatively few pixels), noticeable wave clutter, and low contrast between the boats and the water (comparing figure 4.1 to figure 2.3 shows our data as significantly harsher than many data sets). This means that common features that focus on local detail (such as SIFT) are unlikely to work on our data set. We believe this is appropriate in the context of surveillance; a smaller boat is more likely to be missed by other security measures (human operators, radar, etc.). We include several sequences with larger boats, as the assumption that most of the frame is ocean can lead to spurious salience results if false,

and we believe the data set should reflect this. Indeed, we find that our most promising salience filter has difficulties with detecting large targets.

Our data set has many important features that did not fit into table 4.2, including target out-of-plane rotation (sequences 7, 8 & 18), boats occluding each other (sequences 7, 9 & 10), static salient objects (sequences 10, 11 & 19-22), targets moving out from or in behind clutter (sequences 19 & 20) and ships that break the horizon (sequences 9 & 17). There are many edge cases that can occur that are not present, but we believe the above sequences represent sufficient information to develop an appropriate prior. As we will see this data set is challenging enough that naive approaches will not work.

4.3 Salience Filter

In this section we cover the salience filters we develop for maritime surveillance. We start in section 4.3.1, covering some initial topics that will be relevant to our filters, and highlighting some of the important patterns in the literature. In section 4.3.2 we cover the different filters to be tested. We describe our testing procedure in section 4.3.3, and the results in section 4.3.4.

4.3.1 Overview

Before we can start tracking, we need to be able to remove visual noise from the background. To do this we create a salience filter. Intuitively we want our salience filter to mark any pixels that are likely to be targets and reject pixels that are background. Unfortunately, this is problematic. Should a boat that has stayed in one place long enough be considered background, and how long is long enough? Should white water splashing over rocks be considered salient? Is the white water around a boat part of the boat to be tracked?

This last question is particularly interesting. As humans, with a lifetime of multi-sensory inputs building rich priors, we have a strong idea what a 'thing' is. We know boats to be solid continuous objects, and wake to be a different state of water caused by nearby boats. To an intelligent tracker with only video inputs, boats are segments of contiguous pixels that obey an observation model different to undisturbed water. Wakes can be similarly defined and tend to co-occur with boats. If all one has is video, then the wake is part of the boat in the same way extendible ladders are a part of fire-engines. On a more pragmatic note, wake is strong evidence towards a boat being nearby. Consider a classifier trained to identify images of hammers. Even though a hand is not part of the

hammer, a hand gripping the shaft is supporting evidence for the hypothesis 'hammer'. While the hand is not physically part of the hammer, the concept 'is held by hand' is relevant to the conclusion 'object is a hammer'. Similarly, white water may not be part of the physical boat, but it should be part of what a classifier can use to identify boats. To take this line of thought further, one can ask whether the non-salient water around a boat is part of the boat. Certainly for a classifier, knowing that a patch of salience is fully enclosed by water, and is not part of a larger object, is relevant.

Ideally, we would want our final persistent tracker to be able to explain all the data, assigning each pixel to one of the classes: Boats, Background, Birds, Waves, Miscellaneous Objects, etc. Because our cameras are static, we can use a pixel mask to identify ocean regions, and focus on marking pixels in that region that are not reacting like the larger body of water.

We construct our salience filter as a probability distribution, where the frame produced represents $p(\text{pixel is foreground}|\text{pixel value})$. We are only concerned with the results over the water, however some algorithms would also work against the skyline, or landscape behind the body of water. This stage will act as a pre-filter, with its output being the Y_t used by the rest of the system. Confining this stage to a pre-filter is not necessary. Parameters of the salience filter could be held in M and be included into the SMAE system. We decide not to follow this course of action, as the M we develop in chapter 5 is already large and sufficient performance was attained using our salience filters as pre-filters.

Our literature review covers the salience filters other papers have used in more detail, but we will look at the larger structure in them before moving on to our salience filters. Salience filters tend to construct the background model either by time for each pixel [10, 22, 26, 31–34], or by location for each frame [23, 25, 27, 35–41, 43, 44]. Those that create their model by time keep track of the history for each pixel. They solve some of the spatial problems with the ocean, such as different lighting, water colour, wave texture, or large boats contaminating the model. They look for outliers in time for a specific location. This, however, comes with weaknesses: changes in conditions will make the entire ocean look salient, and objects that stay in a location for long enough will start to be considered background[2]. On the other hand, those salience filters that construct their models by frame solve the problems of time, but have the aforementioned problems of space. These problems are exaggerated in most of the filters that work by frame, as many of them use a static model for each pixel in the frame; thus any frame with a water intensity distribution that is dependent on location in the frame is likely to have problems.

[2]It is debatable whether this is desirable behaviour or not; we believe it is not.

There are three notable exceptions to this. Firstly, Bloisi and Iocchi's [10] static model is a Gaussian mixture model[3]. By modelling many modes, this algorithm will not be led astray by glare such as in sequences 10-12. However, it would have problems on a sequence such as sequence 1, where the boat intensity in one location is the same as the ocean's in another. Secondly, Teusch and Krüger [41] construct their model by pixel row. This will protect against backgrounds that change as they disappear into the horizon, as seen in sequence 17. Lastly, Wei et al. [40] fit a plane to the background using iteratively re-weighted least squares (IRLS), accurately modelling oceans such as in sequence 18, which fades from bright to dark as a function of angle to the sun. We draw attention to this last approach as it forms the basis of our strongest salience filter. By fitting a plane to the image intensity, the salience filter manages to model the large-scale changes in the ocean, without being distracted by small local effects. This unfortunately would not work for instances such as sequence 2, in which the background gets darker in both directions from the center of the frame. However, the idea of fitting a coarse function to the ocean for a given frame will be very useful.

Another observation from the literature is that relatively few measures of 'outlier-ness' are used repeatedly by many of the papers. Noteworthy among these are Gaussian distributions/approximations [10, 22, 32, 39], edge detection algorithms [25, 27, 38], and FFT techniques [42, 43]. These are the approaches we will test. Also worth noting in this context is the work of Bechar et al. [44]. Their salience filter is a combination of several different salience functions:

$$\text{Final Salience} \propto 1 - (\text{Scale-factor}) \prod (\text{Salience filter}), \tag{4.1}$$

where each of the salience filters produces a value close to 0 to indicate salience. This is effectively using a noisy 'OR' to combine different salience measures and is an ad hoc approach. A more appropriate data fusion formula is

$$\text{Final Salience} = \frac{1}{1 + \prod \left(\frac{1}{\text{Salience filter}} - 1 \right)}. \tag{4.2}$$

I have not included the derivation here for conciseness sake; it follows from assuming the sub-filters are independent, and performing inference on them as observations.

The last point we will make before moving on to describe the test salience filters, is an innate problem with background subtraction for adaptive trackers. If a certain part of the target has intensity i_1, and the tracker starts tracking it in front of a background with intensity i_2, then the model will learn the intensity $i_1 - i_2$. If the target moves in

[3] Their salience filter is based on pixel history not by frame, but for argument's sake let us consider a GMM used per frame.

front of a section of background with intensity i_3, the model is now off by $i_3 - i_2$. This will not be a problem for us, because the ocean is largely the same colour.

4.3.2 Selected Salience Filters

In this section, we describe the various salience filters we will test against our data set. The background models will be more accurate with foreground pixels removed. With a tracker operating on the salience filters' results a better estimation of foreground pixels can be achieved, ensuring that the salience filter is neither contaminated by foreground pixels nor discredits background pixels that should be part of the model. With this in mind, we test each algorithm as described (labelled 'XX-1'), and then test it again with the ground-truth guiding which pixels to ignore during the update (labelled 'XX-2') to get an upper-bound on the salience filter's performance with a tracker's assistance.

Filters 1 to 4 are implementations of the most common approaches in the literature. Filter 5 is a novel contribution. The principled fusion of different salient filters (using the sensor fusion equation 4.2) should not be novel, but could not be found in our survey of the current maritime literature. We discuss the works these filters are based on in section 2.2.2. We do not address minimal spatial resolution of these filters as the trackers we will build on top of them require a reasonable resolution.

Salience Filter 1

Salience filter 1 stores a Gaussian history for each pixel by time using its own standard deviation. We use Welford's [51] technique for recursively calculating the mean and standard deviation of each pixel's history. We then consider each pixel's salience to be $1-p(y|history)$. We consider two cases: salience filter 1a uses each pixel's own standard deviation to determine if it is an outlier, while salience filter 1b uses the same standard deviation for all pixels to determine if it is an outlier.

Salience Filter 2

Salience filter 2 calculates the mean and standard deviation for the pixels in the ocean mask for the considered frame and then, similar to filter 1, takes each pixel's salience to be $1 - p(y|frame)$. Isn order to counteract the effect of targets contaminating the sample, we re-weight the data set and repeat the procedure. We consider two cases: salience filter 2a uses each frame's own standard deviation to determine the outliers, and salience filter 2b uses the same standard deviation for all frames.

Salience Filter 3

Salience filter 3 uses the magnitude of the Sobel edge detection algorithm as the salience image.

Salience Filter 4

For salience filter 4 we follow the lead of Sanderson et al. [42]. We divide the image into 32-by-32 pixel sub-windows, calculating the FFT for each. We discard all the DC components, subtract the mean FFT from that of each patch, and inverse FFT the resulting distribution. The resulting image is used as the salience image.

Salience Filter 5

Salience filter 5 is one of our original contributions. It uses a small neural network to try learn the image intensity from pixel co-ordinates. We use a single layer of 12 sigmoid neurons. For input values, we use the set $\{x^a \times y^b\}$ for $a + b \leq 3$. Because the network is so small, it is unable to learn finer details such as targets[4], and creates a good background image. This models the large scale structure of the background (including elements such as glare), while neglecting the smaller details (which tend to be targets). We initialise the network on a sampling of frames from the sequence (which we feel is a fair representation of what the tracker would have had at time t, had it been a persistent tracker), and on presentation of a new frame we let the network update one iteration. In this way, the network can accommodate gradual changes without overreacting to sudden changes. Although we ran this update on every frame, the model does not change often and it could easily be updated less often by another process.

Composite Salience Filters

We also combined the salience results of the above filters[5] using the sensor fusion equation 4.2 to test composite filters. We present these results after the initial 5 filters, labelling each composite filter with a binary string identifier to show which filters were used. Each bit i in the filter name indicates whether filter i was included. For example, 'Comp-10110' uses filters 1, 3 and 4.

[4] The choice of sigmoid rather than Gaussian neurons also contributed to this effect.

[5] We selected 1b and 2b over 1a and 2a due to their better performance, to limit the combinations.

4.3.3 Description of Preliminary Tests

The ideal salience filter would output 1 on any pixel in the ocean mask that is not of the ocean, and 0 for any pixel that is. We allow our salience filters to 'hedge their bets' by assigning fractional answers. An obvious choice for describing a tracker's performance would be precision, recall and F-score. Unfortunately, our labelled data is in the form of bounding-boxes, hence a fair amount of what is labelled in the ground-truth as salient is in fact ocean. This means that the ideal filter would have a limited maximum recall, that could be improved upon only by false positives in the bounding-box. For some targets a large proportion of the bounding-box is ocean, and this distorts the F-score. We settle on using only the precision (i.e. what percentage of the salience allocated falls inside the ground-truth bounding-boxes), knowing that it may favour trackers with a low recall. Because of the ability to 'game' the performance measure, checking the sample frames will be especially relevant.

This choice has an added benefit. The reader will have noticed that the salience filters all lack scaling variables, which would be especially relevant to interpreting 'soft' salience answers. If we are simply measuring the fraction of the awarded salience that falls inside the bounding-boxes then scaling factors fall away, leaving us with one fewer independent variable over which to optimise.

Our measure for performance on a sequence is therefore

$$score(\text{Salience Filter},\ \text{Sequence}) = \frac{\sum_{frames} \text{Salience inside targets}}{\sum_{frames} \text{All salience}}. \tag{4.3}$$

In order to get a score for a filter across the sequences, we need to decide whether to weight the sequences according to their frame count or not. We feel that the sequences represent different conditions, and so their score represents the performance in different use cases. Weighting a particular case heavily just because we have more frames of it is undesirable. Thus we allocate to a filter the score

$$score(\text{Salience Filter}) = \text{average}(score(\text{Salience Filter}, \text{Sequence})). \tag{4.4}$$

We find that on the data this measure does not match our intuitive response for ranking the different filters. The results using a harmonic mean match our intuitions better than with the arithmetic mean. We will discuss this in our presentation of our results.

We run each salience filter on the entire data set twice to evaluate both its naive performance[6] and its upper-bound performance[7].

4.3.4 Results of Preliminary Tests

Figures 4.3 and 4.4 show sample frames for the salience filters for the naive case and the upper-bound respectively. Each row shows one of the filters, and each column its response for a particular sequence. Noteworthy features in these images are as follows: the target in sequence 14 has a salient patch behind it for Filter 1a-1 and 1b-1, where a slow-moving boat has contaminated the background model, leading to ocean registering as salient; filters 1a-2 and 1b-2's rectangles of solid white (salience), caused by an edge case involving regions that are salient for the entire sequence (static objects); filters 2a-1 and 2b-1's large swathes of false positive due to non-uniform ocean lighting; filter 3-1's large quantities of noise, and only catching the borders of salient objects; and filter 5-1's failure to catch the large ship (if a boat is large enough, it can be worth the neural network dedicating its limited resources to modelling the boat). This is corrected in filter 5-2, suggesting that feedback from the tracker will improve filter 5's performance on targets. These images largely match the rankings shown in figure 4.5(B) (to be discussed below). Sample frames for more sequences are available in appendix A, and at `http://www.dip.ee.uct.ac.za/~cbradshaw/PhD_data/`.

[6] That is, without the tracker providing feedback to assist in generating the background model.
[7] Providing feedback from the ground-truth to keep the background model accurate.

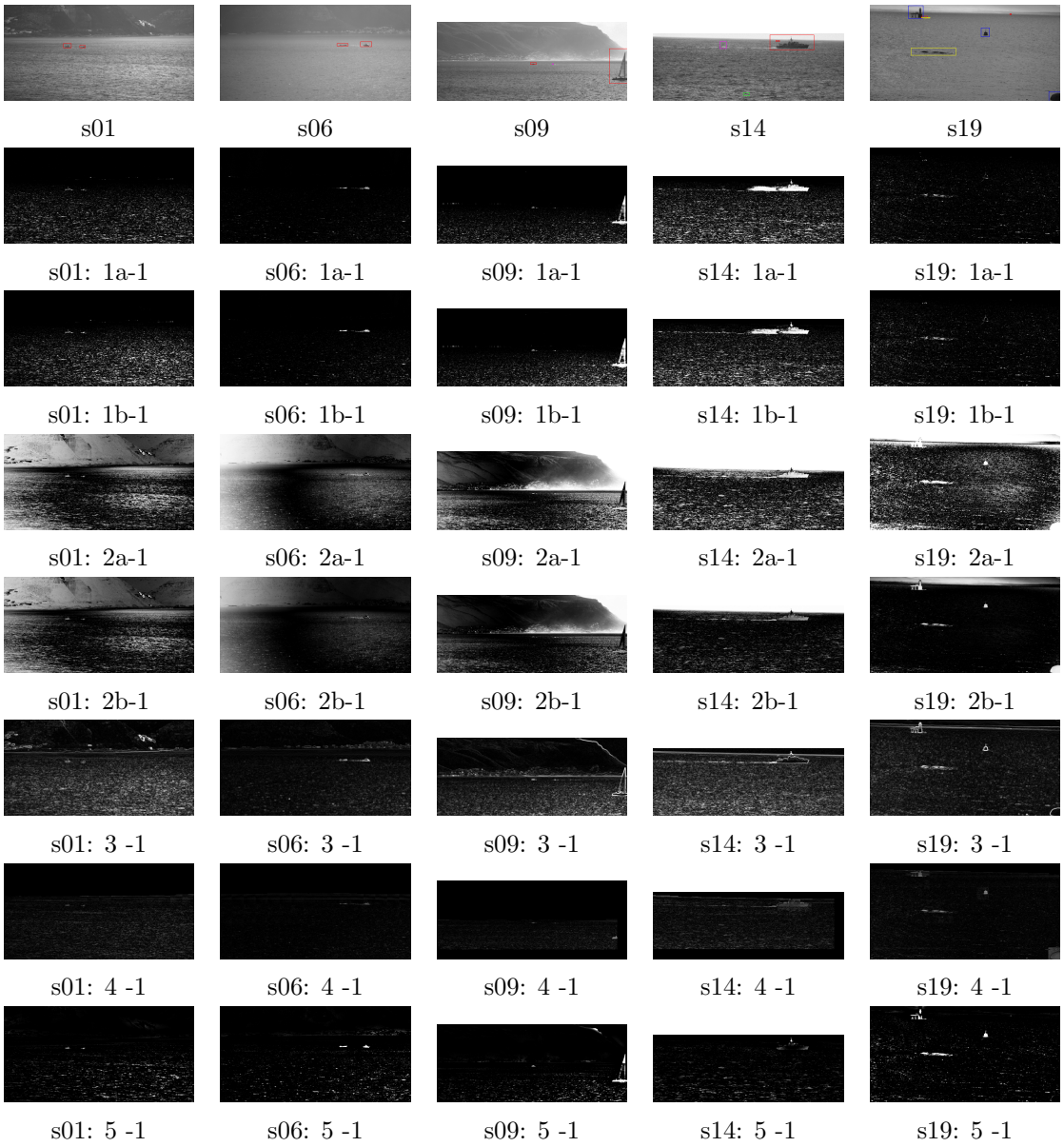


FIGURE 4.3: Sample salience results for naive filters.

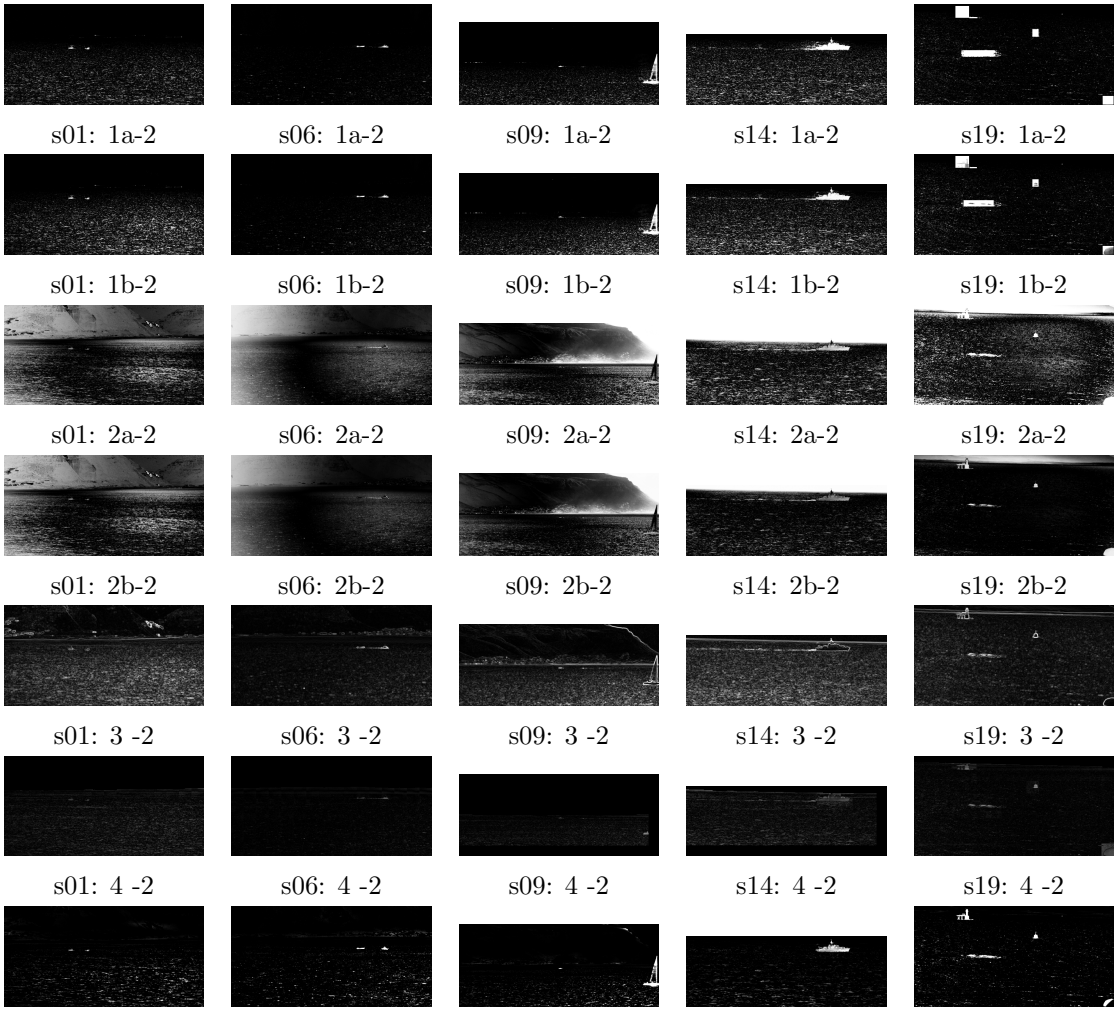
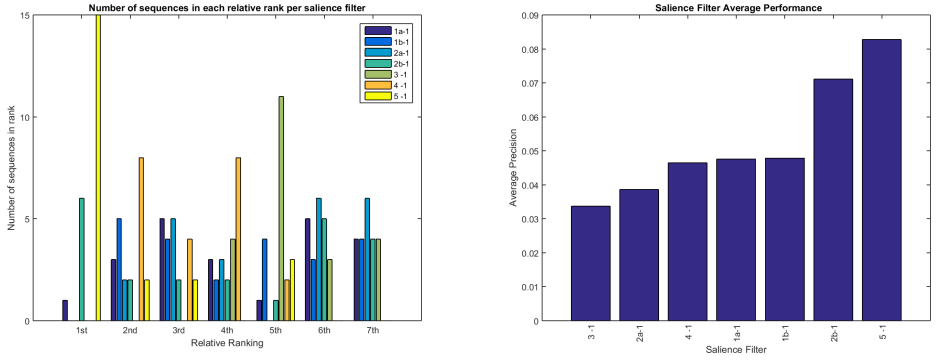


FIGURE 4.4: Sample salience results for upper-bound filters.



(A) (B)

FIGURE 4.5: Salience filter performance: (A) shows the distribution of relative ranks each filter achieved in number of sequences. (B) shows the average precision for each salience filter across all sequences.

Figure 4.5(A) shows the distribution on relative ranks between the salience filters. For each sequence, the salience filters are ranked according to their average precision, and the number of sequences in each rank are totalled. While it is clear that filter 5 dominates on most sequences, the relative performances do fluctuate. Figure 4.5(B) shows the average performance of each of these filters. Here we can see that filter 5 is the most successful. As mentioned in section 4.3.3, these results should be seen in the light of figures 4.3 and 4.4, which confirm that filter 5 is not 'gaming' the performance measure.

Figure 4.6 shows the performance change between naive implementation of each of the initial filters, and the upper-bound implementation. Most of the classifiers are close to their upper-bound, except for filter 1 which achieves drastic improvement. These are largely due to the edge case that occurs when a patch has been labelled as foreground for an entire sequence, as mentioned regarding figure 4.4. This gives an unrealistic boost to the precision. Note that even with this behaviour, the upper-bound performance is not much better than that of filter 5.

Figure 4.8 shows the performance for all the composite filters, and figure 4.7 shows sample frames for some composite filters (each row shows progressively stronger filters according to their average). From the average performance it seems as though adding more filters into a composite generally improves it. However, looking at the sample frames tells a different story. For the first few rows the composite filters are better than their component filters, but by the last two rows the filters are taking advantage of the precision measure: committing to very few pixels that are certain, missing many salient pixels, but still getting a high precision score. Indeed, the best composite Comp-011111 hardly gives any response on sequence 1, and leaves out the entire sail in sequence 9. In light of this, we present figure 4.9, which uses the harmonic mean rather than the arithmetic mean to combine the sequence scores. The harmonic mean is closer to the

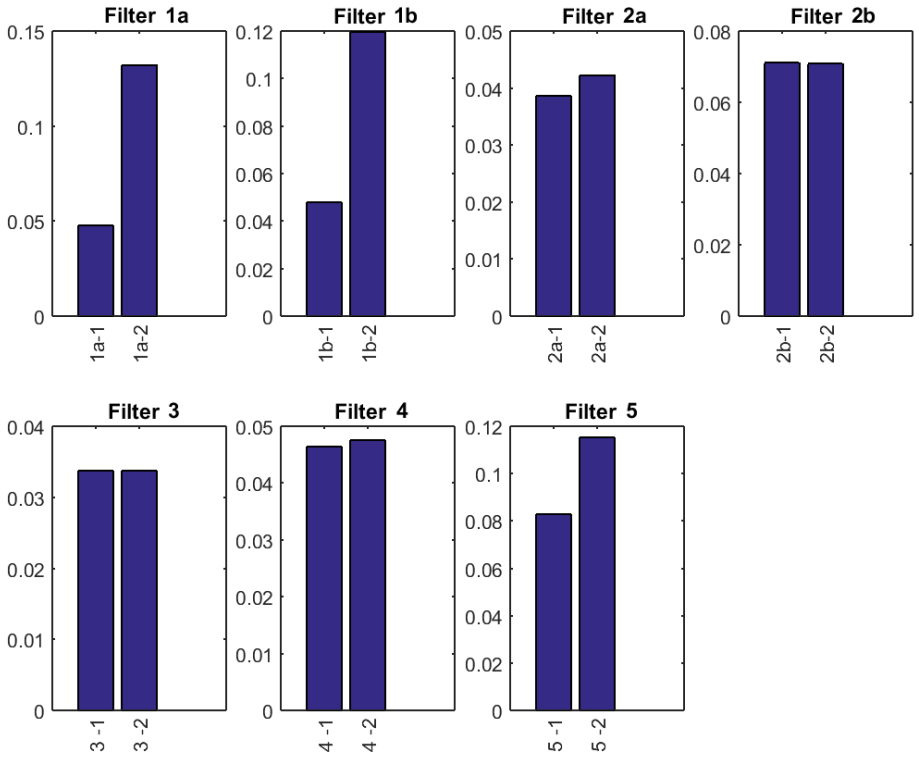


FIGURE 4.6: Naive and upper-bound performance for each salience filter.

minimum of a set than the arithmetic mean, and so punishes filters that get all their 'mass' in a few sequences and rewards those that score consistently. This does not solve the problem of filters not committing to pixels inside a sequence, but does punish those like Comp-01111 that write off entire sequences. Now our rankings compare to what we see in the sample images. It is interesting to note that even though filters 1-4 (shown in Comp-10000, Comp-01000, Comp-00100 and Comp-00010) change ordering relative to each other, they still perform worse than filter 5[8].

[8]We refrain from showing figure 4.5(B) again with the arithmetic mean replaced by the harmonic mean, as its results are embedded in figure 4.9.

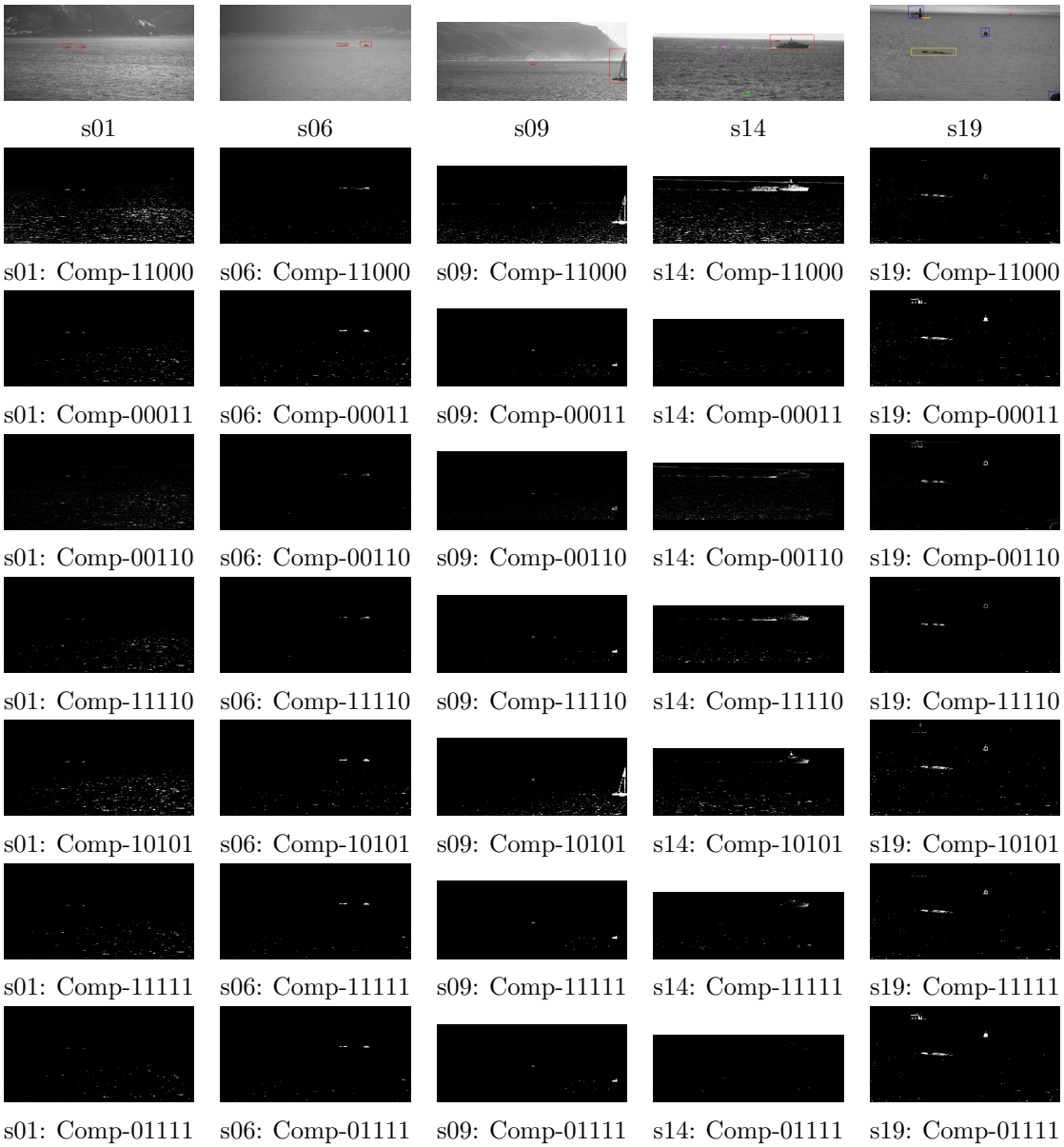


FIGURE 4.7: Sample salience results for composite filters.

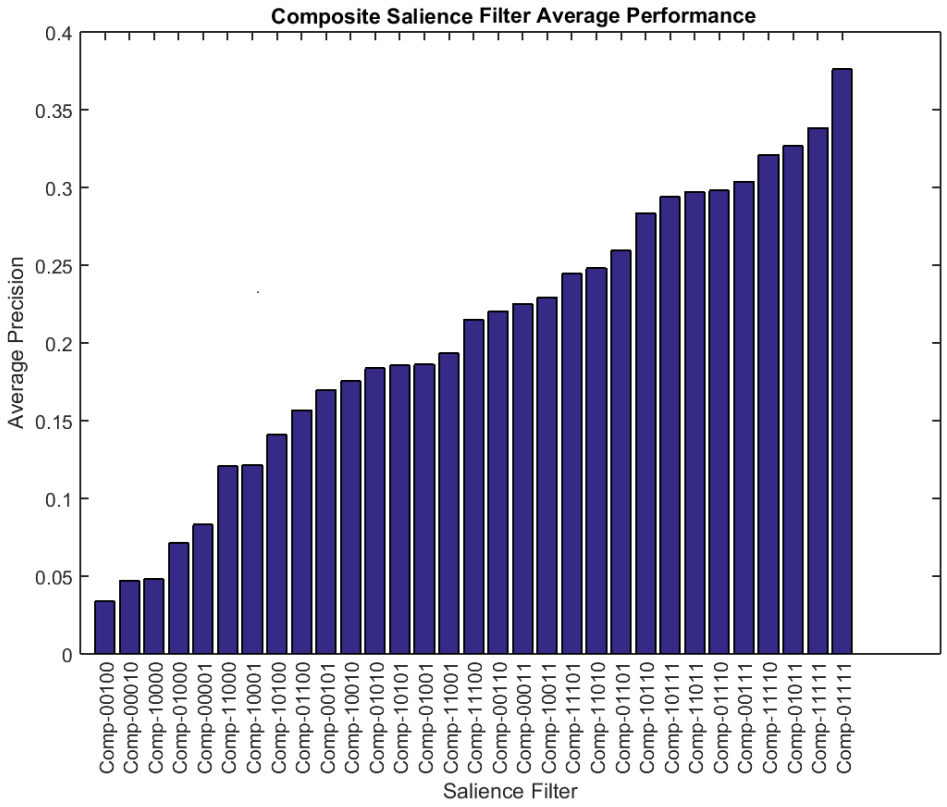


FIGURE 4.8: The average precision for each composite filter.

We remind the reader that the salience filter is not our major contribution, but rather a pre-filter that we address to facilitate the application of our framework to maritime surveillance. The contributions in this chapter are listed to in section 1.6, but the main contributions of our work are in the subsequent chapters. Giving precisions without recall does not produce convincing results. We deemed the work involved in accurately labelling enough data to get representative recall values unwarranted, given that this section is a merely a stepping stone on the path to the application of SMAE in chapter 5. We found that while incorporating multiple salience filters generally lead to better results, it was not universal.

In light of these results, we proceed with filter 5 (our best performing single filter), which achieved high precision and produced reasonable sample images. It may have struggled on large targets, but the sample image for sequence 14 shows a significant improvement between naive and upper-bound cases on large targets for filter 5. We believe that using the tracker output to label which data is used for the neural network training should show improvements for the filtering of large objects.

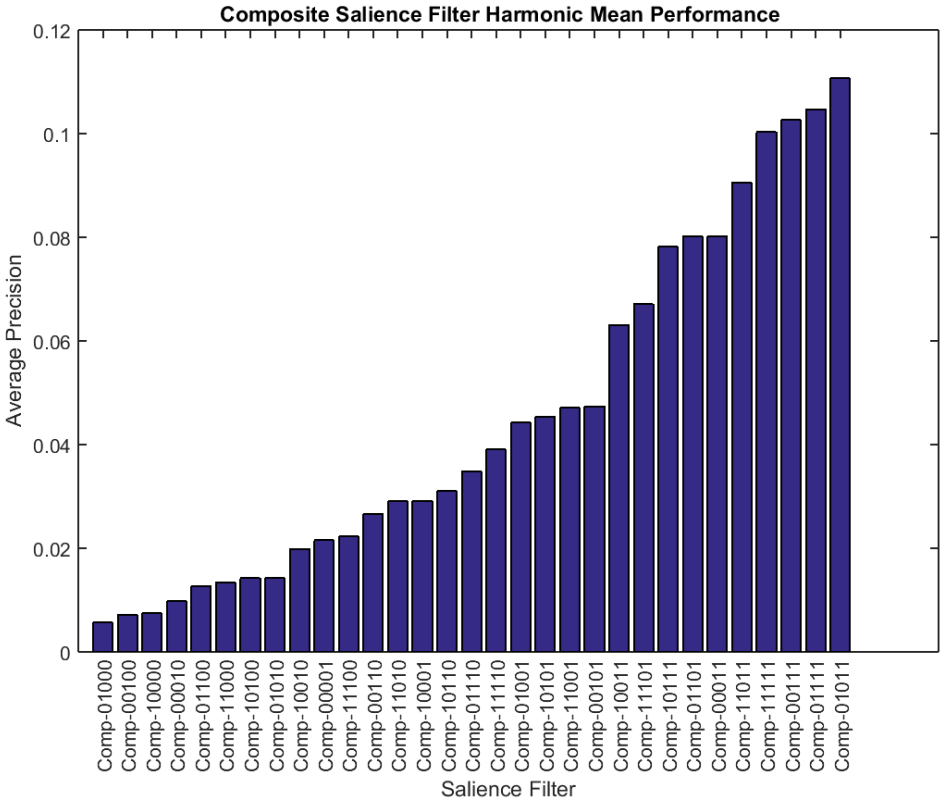


FIGURE 4.9: The harmonic mean precision for each composite filter.

Chapter 5

Maritime Surveillance with SMAE

This chapter continues our application in maritime surveillance. In the previous chapter, we described our data set and presented the salience filter that acts as a pre-filter for our framework. In this chapter we apply SMAE to the task of maritime surveillance. In section 5.1 we develop an adaptive tracker; that is, an application of SMAE that will handle single adaptive target tracks, holding both target state and target model in a joint distribution. In section 5.2 we extend this to a persistent tracker; i.e. one that improves its tracking performance over time, learning from past tracks. We see our tracker as a nascent intelligent agent, thus our goal is not to create an optimised working system, but rather to create the conditions that lead to accurate Bayesian learning. Finally, we discuss the extension of this framework to more common features. It is worth mentioning that this chapter is intimately linked with chapter 3, as it is implementing the multiple target VOT tracker derived therein.

5.1 Adaptive Tracker

In this section we describe the adaptive tracker to be used in our MTT. The goal of the adaptive tracker is to track a target for as long as possible once initialised from the salience filter. Our priority in designing the adaptive tracker is in creating a good foundation for the persistent tracker, rather than in creating a strong tracker. We attach more importance to how easily the tracker is bootstrapped than to the strength of its initial performance. This section proceeds as follows: section 5.1.1 covers the adaptive trackers we test and how they relate to our derivation, section 5.1.2 describes the tests done on these trackers, and section 5.1.3 presents the results of these tests.

5.1.1 Base Adaptive Trackers

While it would be preferable to test our systems against other existing algorithms, we found none for which the assumptions were close enough to ours to lead to a fair comparison. Those systems in the existing literature that had similar use cases (self-initiated trackers with static cameras mounted high above the plane of water) use large amounts of prior information to build strong vessel classifiers [10, 16, 43][1]. The focus of our system is on bootstrapping those priors from the little available information, and so testing against these systems would bias the tests against our system. On the other hand, testing against standard VOT tracking algorithms would face the opposite problem: our system has been designed with the noise of ocean backgrounds in mind, and so would have an advantage. We decided that the effort required in adapting any current solution to be a fair baseline comparison is not justified, and so we consider our tests on the adaptive tracker's results as a baseline against which we can compare our persistent tracking results in section 5.2. We now this is not ideal, however running comparisons based "unfair" test conditions would not produce meaningful results.

Our adaptive tracker is a monolithic Bayesian system that takes the output of the salience filter as its observations Y_t, and makes inferences on the posterior probability of a target of interest being at locations in the joint (M, S_t) space. The difference between this and the standard Bayesian adaptive tracker is illustrated in figure 1.3. To do this, it employs an adapted particle filter, where each particle represents a discrete value of all variables in S_t, but holds a complete representation of the PDF for the hyper-plane that stretches across all the variables in M. Figure 5.1 illustrates this for the VOT case, in comparison to the simple 1-dimensional case shown in figure 3.7, highlighting the variables stored in a single particle. We will not go over the basics of particle filters due to space reasons, and refer the reader back to chapter 3 for justification of our particle filter framework.

In section 3.3.3 we discussed that when particles overlap we need to handle the inference for the various cases separately. We do this to ensure that every hypothesis our tracker considers is required to explain all the evidence in a frame; practically, this means updating the information in M correctly according to the different allocations. Our solution in chapter 3 was to group particles into clusters. Particles in a cluster must co-occur, and each cluster maintains a list of clusters with which it cannot occur. Whenever two proposed clusters (C_1 and C_2) overlap, they are replaced by three clusters: the

[1] This may seem like a small set of viable papers in comparison to the wealth of contributions mentioned in chapter 2. It excludes those which are classification papers [18, 22, 41, 45, 52], those that are for different surveillance modes [23–25, 27–29, 39, 46, 47, 49, 50], those that are not adaptive trackers [32, 33, 40], and those with little or no quantitative testing [1, 26, 30, 31, 34–38, 42, 44], which we interpret as exploratory papers. Those listed in the text are the best fitting examples from the literature, yet even they are not fair comparisons.

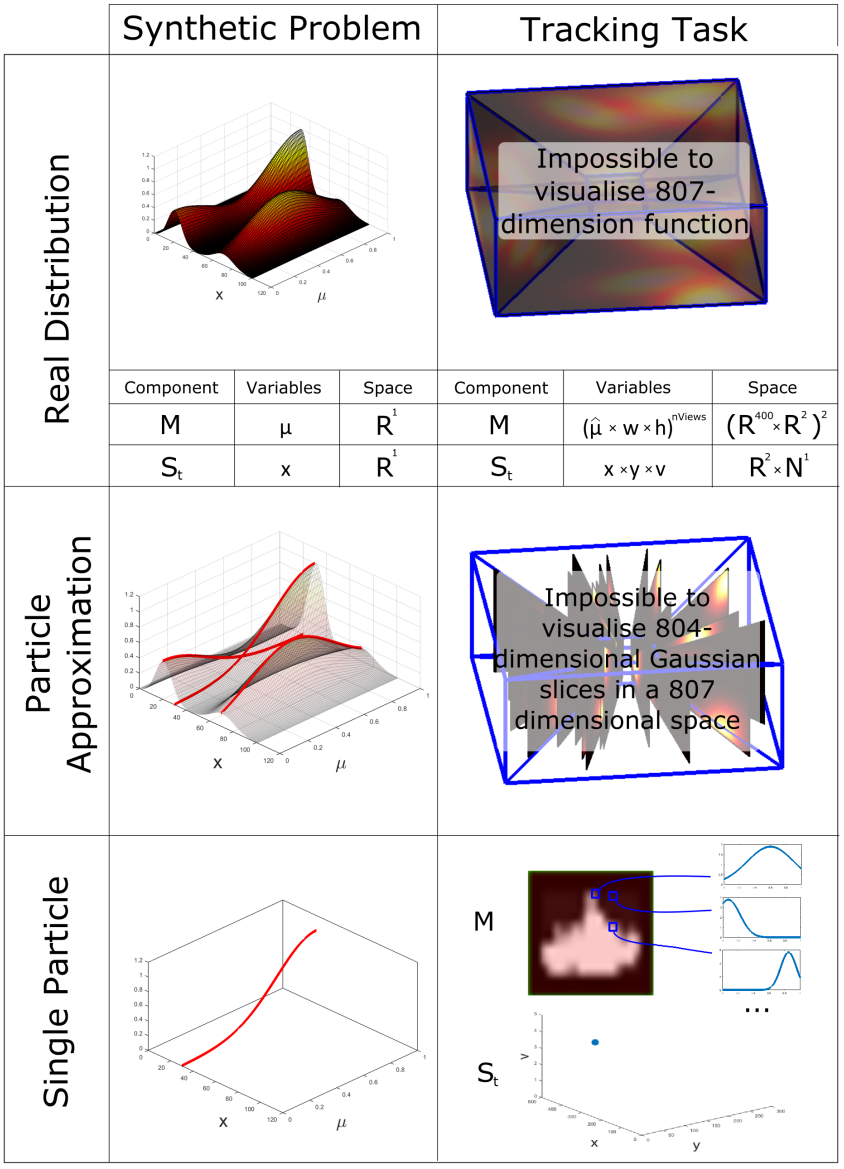


FIGURE 5.1: The adapted particle filter as applied to the synthetic problem and the tracking task. The first row shows an example of the PDF to be tracked, summarising the model and state components. For the synthetic problem, the model is defined by a single variable μ, the center of the observation model, and the state is defined by the single variable x. For the tracking task, the model is joint across a number of views (we used 2), where each view contains a grid of distributions (we used a square of side 20 pixels) each representing the mean of a Gaussian observation model for that location, and a width and height for the view. The state is a vector containing x, y, and view index. The second row shows our adapted particle filter, where each particle is discrete in all the state variables, but holds a distribution across the model variables. The final row shows a single particle; the synthetic problem's particles can be stored as the state variable and parameters of the Gaussian distribution on μ, i.e. $(x, \bar{\mu}, \sigma_\mu)$. For the tracking task, we need to store the state and the parameters of all the distributions across the observation grid and size parameters, i.e. $(x, y, v, (\bar{\hat{\mu}}, \hat{\sigma}_\mu, \bar{w}, \sigma_w, \bar{h}, \sigma_h)^{nViews})$.

union which covers the case that both are present, and updates each M according to the allocations from equation 3.23; and one for each cluster covering the case that the other is not present, where the present cluster's M is updated having the full weight of pixel allocations. We refer the reader back to figure 3.10 to visualise the process of generating the new clusters.

At each time instant, we include two new clusters centred on the largest (normalised for perspective) patches of salience, with the set of clusters predicted from the previous time instant. We consider the LLR for each cluster (using equation 3.14 with equation 3.23 handling pixel overlaps) taken without considering the other clusters to be indicative of the local fit of the cluster. We perform a gradient descent on the positions of the particles so as to maximise this LLR. This gradient descent will position the particles such that they have a maximal LLR, but will also update the M for each particle to hold the correctly inferred information for that configuration. This takes into account pixel overlaps using equation 3.23 to perform the optimal approximation for inference on the different particles' M values for different pixel allocations. During this gradient descent, cluster formations will change as particles overlap and separate, joining and splitting clusters respectively. Once this gradient descent with cluster updating is finished, we calculate the posterior on each particle, marginalised over the valid subsets of particles (as dictated by the cluster properties).

The information stored in M is large, and the inference is a relatively heavy operation; this leads to an impractical overhead when the number of particles is not small. We found that the gradient descent made using fewer particles effective. This means that considering all valid subsets of particles, which would be infeasible with a more standard number of particles, is now feasible. Figure 5.2 walks through the application of this algorithm to a sample frame (note the similarity to figure 3.11).

We test three adaptive MTTs using the different observation models described in section 3.5.1: a Gaussian distribution centred at $\vec{\mu}$ with a standard deviation σ_{obs} of 0.2, where $\vec{\mu}$ is a parameter being inferred; an alpha mask in which the pixel value has an α chance of being drawn from a uniform distribution, otherwise is drawn from the background model, with $\vec{\alpha}$ as the parameter being inferred; and a combination, where the pixel has a α chance of being drawn from a Gaussian centred at μ and both $\vec{\alpha}$ and $\vec{\mu}$ are inferred.

For each of these cases, a grid of parametrised distributions is stored (as described in section 3.5.1), with the cropped portion of each observation transformed into the particular grid's scale using the techniques developed in section 3.5.2. Because the observation model is in the form of a LLR, using just the cropped patch is consistent with considering the entire frame as the observation Y.

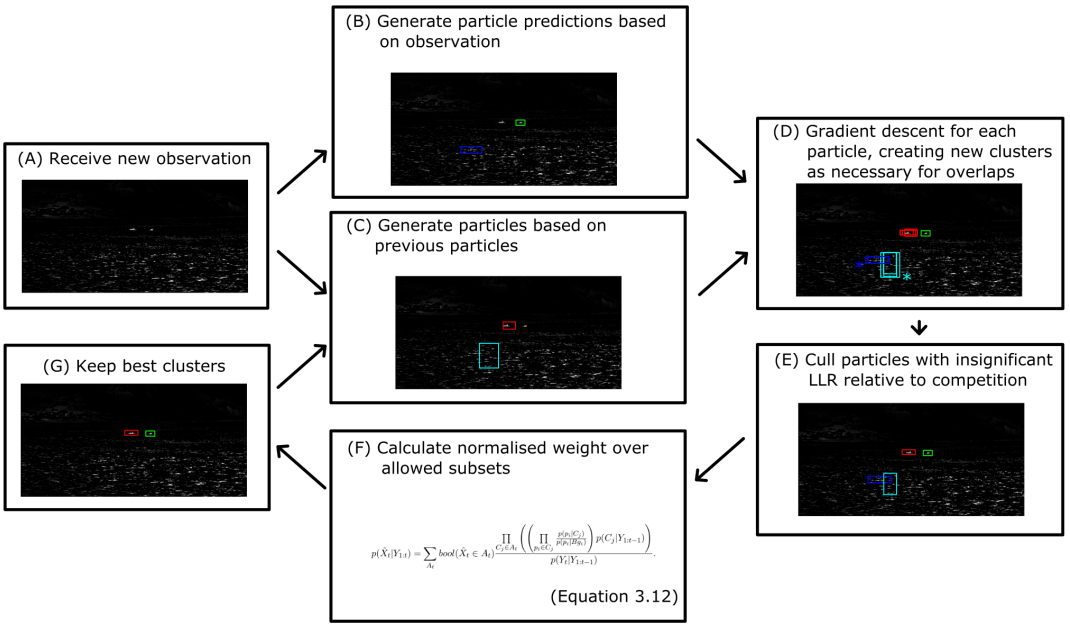


FIGURE 5.2: A single iteration of the tracking algorithm. For each frame that is received (A), a set of hypothesised particles is created using the large regions of salience (with perspective taken into account) (B) and the previous frame's output (C). These particles are then fitted using a gradient descent optimising the LLR (D), creating new clusters as necessary. In this example, the blue and cyan particles overlap, and so the algorithm creates a cluster for the combination of particles in addition to the two original clusters (indicated with asterisks). The algorithm then culls those particles whose LLR is insignificant compared to those with which they compete (E). In this case, that is the clusters involved with the blue and cyan particles; the combination cluster has a much larger LLR, and the two individual clusters are rejected. Finally, the algorithm uses equation 3.12 to calculate the posterior for each particle (F), and keeps the best ones for the next frame (G). For all the frames, only the bounding-boxes are shown. Each particle has a distribution in M (not shown), as illustrated in figure 5.1.

We draw attention again to the fact that this is a MTT. The adapted particle filter described in section 3.3.2 models the different particles as hypothesised single targets, and keeps track of which particles are considered compatible and which are considered incompatible. In this way, the set of particles encapsulates the joint multi-target PDF.

The purpose of these tests is two-fold: firstly, to establish which observation model to take further with our tests, and secondly, to establish a baseline against which to test our persistent tracker's improvements.

5.1.2 Description of Tests

As described in section 4.2, our data set consists of 22 maritime sequences taken from a static camera containing many different types of salient objects. For each sequence, we

run each of our base multi-target trackers, and record the following for the detections in each frame: the bounding-boxes, the distribution on M, and the posterior weight.

Deciding on a performance measure is difficult for any MTT. While it is easy to define what the ideal output would be, the many possible failure modes make it challenging to design a scoring system that does not reward the performance of some undesirable edge case. We let our use case guide us in our decisions regarding the metric.

There are two common measures used for describing how well a single target tracker matches its underlying target. The first is the ratio of the intersections of the bounding-boxes to the union of the bounding-boxes[2], and the second is the difference between the two bounding-boxes' centroids. In our case we do not want to be too prescriptive on the bounding-box. Our interests are in the presence of boats, their approximate location, and inferring accurate appearance models. To this end, we set a threshold on the overlap between tracker and ground-truth of 0.15, above which we consider it a hit. Figure 5.3 shows an example of overlaps to justify a threshold this low.

FIGURE 5.3: Two bounding-boxes that we would consider valid that have an overlap score of 0.15.

Regarding the scoring of the assignments, we take our lead from Smith et al. [19]. Their nine metrics are discussed in the literature review; we ignore the CD rate as its information is carried better in the MO and MT rates. The FIT, FIO, TP, and OP measures all address the issues around tracks switching targets. They are predicated on the assumption that each target should be tracked by one track and each track should track one target. We discussed the issues involved with this in section 2.1 with figure 2.1. We will use the SW measure defined at the end of this section as a measure of how often the best track following a target switches.

[2]Equivalently, Dice's coefficient.

Calculating the MO, MT and SW measures becomes intricate when considering different thresholds for our posterior probability. In our context, they all relate to a failure to keep the right particles. On account of this, we calculate them with a threshold of zero: any particle that is part of the tracker's output contributes to these measures. We also consider the tracker's precision (P) and recall (R) for different thresholds θ and combine them into the F-score (F) for a tracker. Thus our metrics for a given tracker on a given sequence are[3]:

- $\text{MO(sequence)} = \frac{1}{\text{nFrames}} \sum_{frames} \frac{\sum_{\text{trackers on target}} (\text{number of targets best described by tracker} - 1)}{\max(\text{number of visible targets}, 1)}$
- $\text{MT(sequence)} = \frac{1}{\text{nFrames}} \sum_{frames} \frac{\sum_{\text{tracked targets}} (\text{number of tracks assigned to target} - 1)}{\max(\text{number of visible targets}, 1)}$
- $\text{SW(sequence)} = \frac{1}{\text{nTargets}} \sum_{targets} \frac{(\text{number of trackers assigned to target} - 1)}{\text{number of visible frames}}$
- $\text{P(sequence, } \theta) = \frac{\sum_{frames} (\text{number of track responses on target with posterior} > \theta)}{\text{number of tracks with posterior} > \theta}$
- $\text{R(sequence, } \theta) = \frac{\sum_{frames} (\text{number of targets tracked with posterior} > \theta)}{\text{number of targets}}$
- $\text{F(sequence, } \theta) = \left(\frac{\text{P(tracker, sequence, } \theta)^{-1} + \text{R(tracker, sequence, } \theta)^{-1}}{2} \right)^{-1}$

We also need to define an aggregation on these scores in order to get a representative value for a tracker across all sequences. We follow our decision from section 4.3.3 to use the harmonic mean. The harmonic mean favours distributions that are more consistent over those with a large spread. We make this decision as the quantitative results it produces are closer to the qualitative results we will present alongside. In appendix B, we present the results using both the harmonic mean and the arithmetic mean, that the reader may be satisfied with this unusual choice.

We also show a selection of learned observation models for each tracker, to give a qualitative appreciation of whether the appearance model is converging correctly or not.

5.1.3 Results of Tests

Figure 5.4 shows the quantitative results for the three adaptive trackers. The top graph shows the PR 'curve' for trackers (as the thresholding will be a result of $p(I|M, Y_{-\infty:t})$, the PR curves are constrained to a single point) as an aggregated value across all sequences. These points may seem to represent poor performance. We point out that

[3]Where MO, MT, and SW stand for multiple object, multiple tracks, and switches respectively.

the harmonic mean produces lower values than the arithmetic, and that our labelling includes objects that are unlikely to be detected (to leave space for improving trackers). The lower left bar graph shows the average MO, MT and SW errors for each tracker, and the lower right graph shows the number of sequences each tracker ranked in each position (as judged by the best F-Score). The only measure in which the tracker with the alpha-mask was not the best was for the MT error. We are not overly concerned about the MT errors; they are mostly caused by trackers locking onto different parts of large vessels. As our primary concern is smaller vessels, this is not a relevant issue.

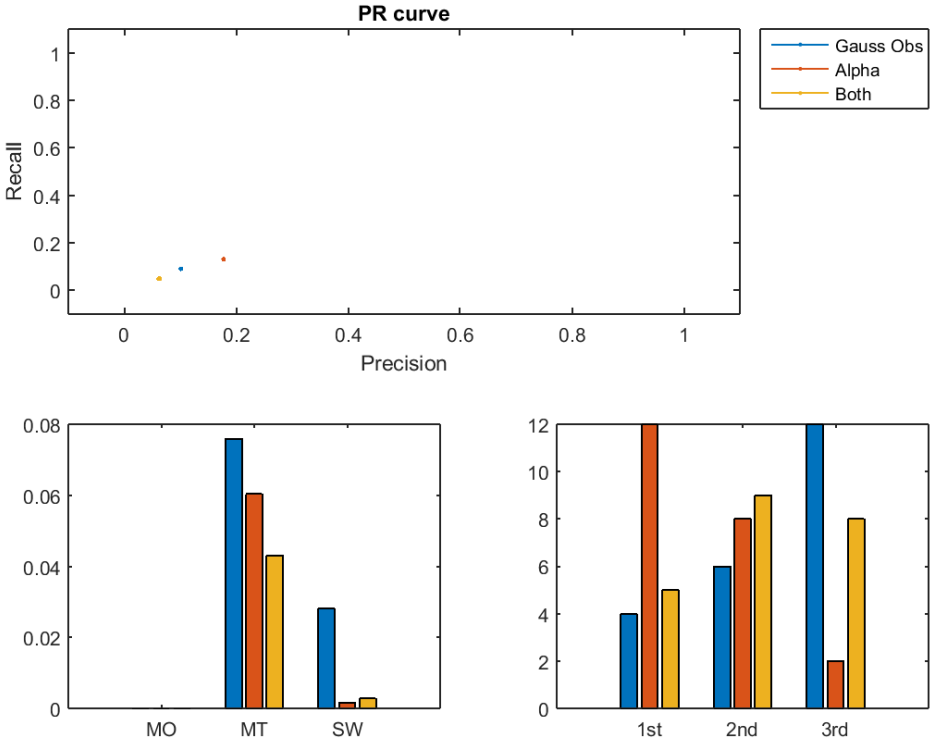
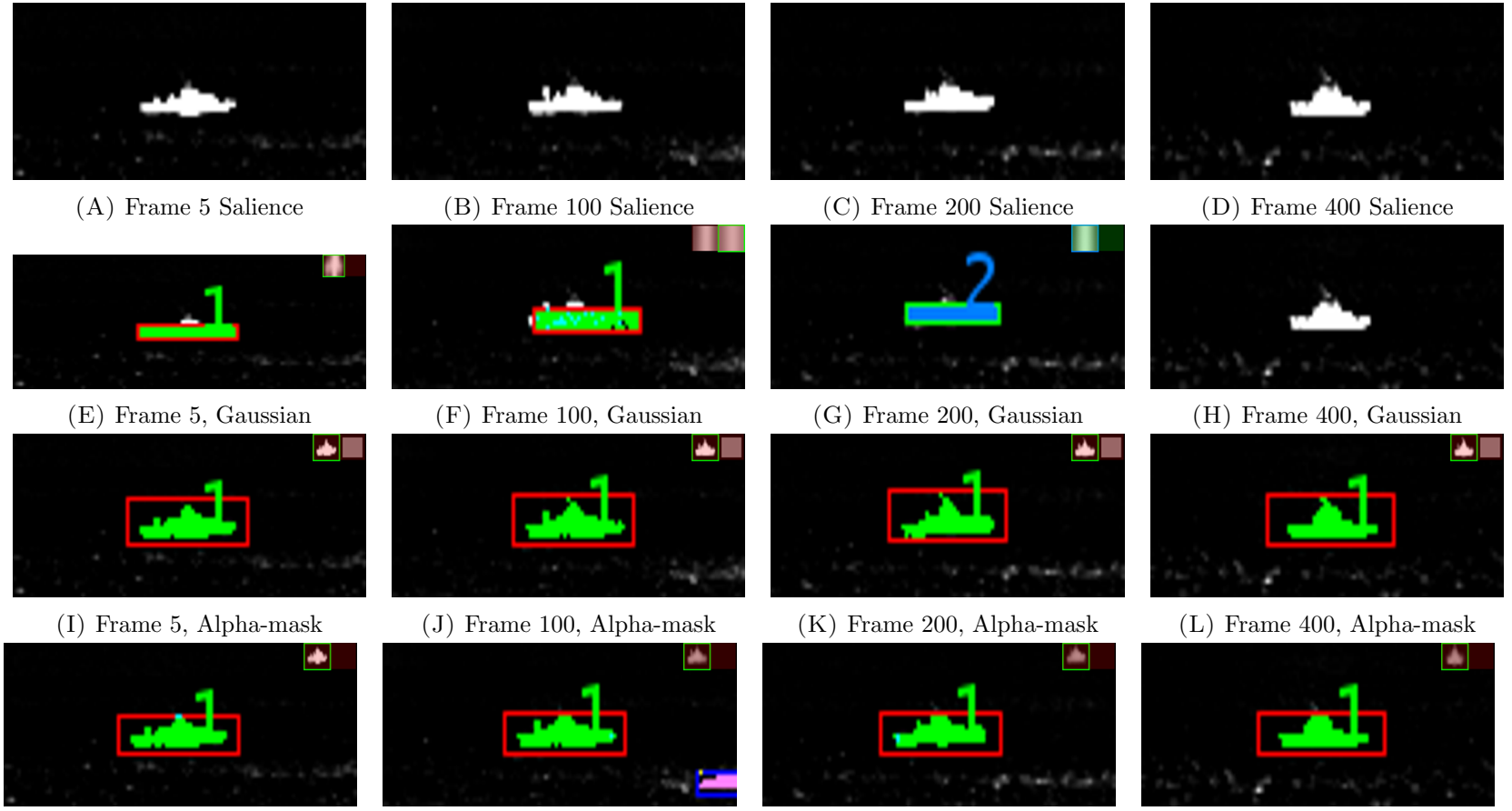


FIGURE 5.4: The top shows PR points for the three adaptive trackers. Learning to reject tracks, and hence move along a PR curve, comes with training that will be integral to our persistent trackers. The lower left shows MT, MO, and SW scores for the different trackers. The lower right shows the number of sequences each tracker ranked in each position (as judged by F-score). For the MT, MO, and SW errors we show the arithmetic mean instead of the harmonic, as the harmonic mean of any set with a zero element is zero, and hence would be meaningless for these metrics.

We also include samples of the templates used by the trackers for different targets in figures 5.5, 5.6, 5.7, and 5.8. For each target, we show the cropped target at several frames in the salience filter and in each tracker. For the tracker images, we insert the templates into the top right of the image. The bounding-box in each frame is the colour of the cluster with which it is associated, and any pixels the tracker decided originated from a target are coloured with its colour (along with the particle number

for that target). In the insert, the two views' templates are shown (μ for the Gaussian observation, and α for the other two observation models). The templates are shaded the colour of the cluster to which they belong, and the visible template is bordered in the target's colour.

The cluster colour changes from frame to frame, hence the bounding-box's colour is unimportant. However, a change in the pixel colours indicates a change in the underlying target identity, and is hence a SW error. The tracker with a Gaussian observation distribution has a noticeable blur in all its templates, which exists to a lesser extent in the tracker with both an alpha mask and a Gaussian distribution. The tracker with only an alpha mask learns templates with much sharper boundaries. The problems associated with the larger objects stem from the salience filter accommodating them into the background model, as can be seen in the top row in figure 5.8. A boat that has been included into the background will generate smaller salient regions on its peripheries. For the small target in sequence 9, the alpha-masked tracker tracks until the object is obscured following frame 200, then reinitialises. The entire video set for each tracker is available at `http://www.dip.ee.uct.ac.za/~cbradshaw/PhD_data/`.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

(E) Frame 5, Gaussian (F) Frame 100, Gaussian (G) Frame 200, Gaussian (H) Frame 400, Gaussian

(I) Frame 5, Alpha-mask (J) Frame 100, Alpha-mask (K) Frame 200, Alpha-mask (L) Frame 400, Alpha-mask

(M) Frame 5, Both (N) Frame 100, Both (O) Frame 200, Both (P) Frame 400, Both

FIGURE 5.5: Sample templates for a medium-sized target in sequence 6 for trackers with different observation models with no persistence.

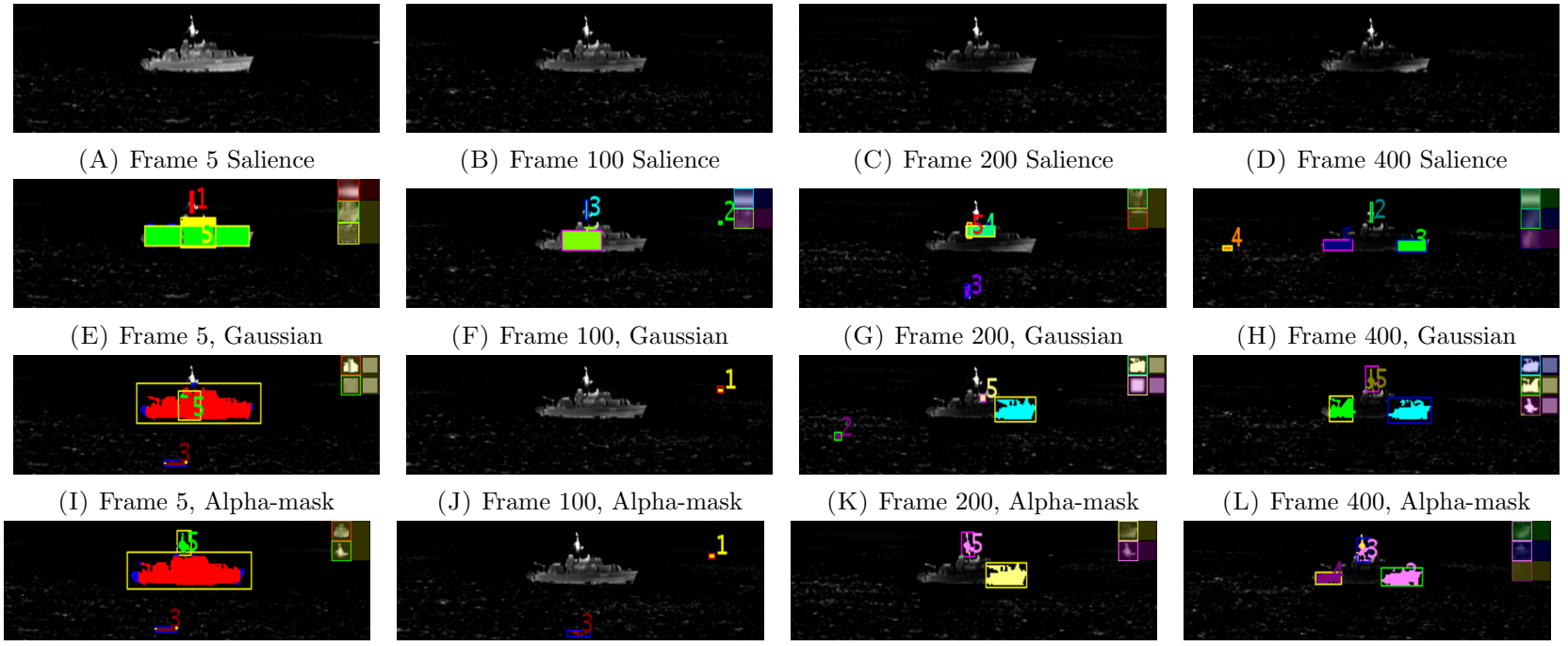
(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

(E) Frame 5, Gaussian (F) Frame 100, Gaussian (G) Frame 200, Gaussian (H) Frame 400, Gaussian

(I) Frame 5, Alpha-mask (J) Frame 100, Alpha-mask (K) Frame 200, Alpha-mask (L) Frame 400, Alpha-mask

(M) Frame 5, Both (N) Frame 100, Both (O) Frame 200, Both (P) Frame 400, Both

FIGURE 5.6: Sample templates for a small target in sequence 9 for trackers with different observation models with no persistence.



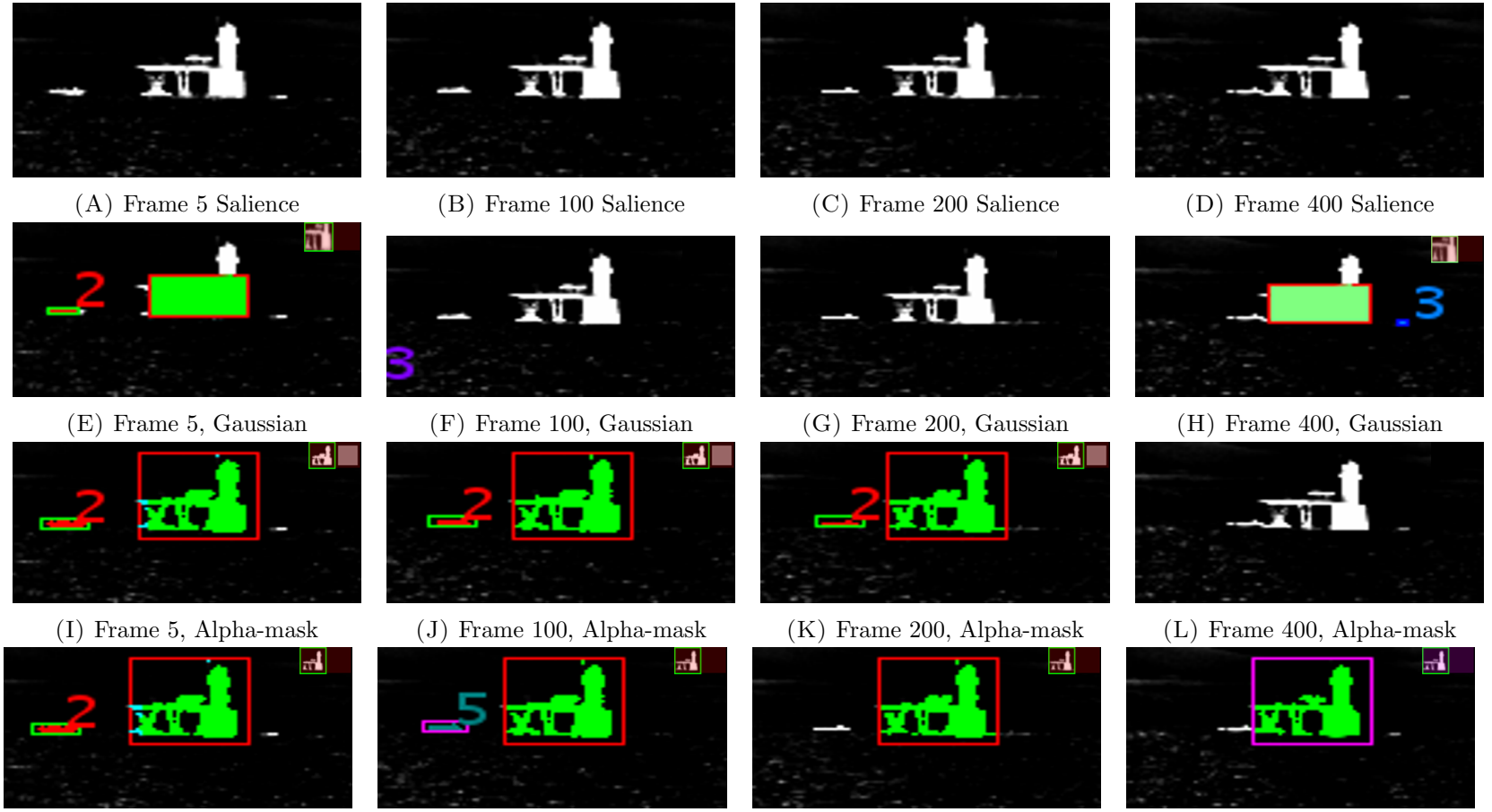
(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

(E) Frame 5, Gaussian (F) Frame 100, Gaussian (G) Frame 200, Gaussian (H) Frame 400, Gaussian

(I) Frame 5, Alpha-mask (J) Frame 100, Alpha-mask (K) Frame 200, Alpha-mask (L) Frame 400, Alpha-mask

(M) Frame 5, Both (N) Frame 100, Both (O) Frame 200, Both (P) Frame 400, Both

FIGURE 5.7: Sample templates for a large target in sequence 14 for trackers with different observation models with no persistence.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

(E) Frame 5, Gaussian (F) Frame 100, Gaussian (G) Frame 200, Gaussian (H) Frame 400, Gaussian

(I) Frame 5, Alpha-mask (J) Frame 100, Alpha-mask (K) Frame 200, Alpha-mask (L) Frame 400, Alpha-mask

(M) Frame 5, Both (N) Frame 100, Both (O) Frame 200, Both (P) Frame 400, Both

FIGURE 5.8: Sample templates for a static object in sequence 19 for trackers with different observation models with no persistence.

We reiterate that the adaptive tracker is not meant to produce strong results. Our intention was to create a baseline tracker on which to implement our persistent learning. Based on both the quantitative performance results and the qualitative sample templates, we proceed with the alpha-masked uniform distribution as the observation model for our persistent tracker.

5.2 Persistent Tracker

Having designed and tested our adaptive trackers, we extend our adaptive tracker to be a persistent tracker — one designed to run over extended periods of time and progressively improve its performance. Our goal is to extend the framework around our adaptive tracker (which only adapts its observation model inside each track) to a tracker which updates the prior it carries into each track. In section 5.2.1, we show how to extend the adaptive tracker, and present the machine learning framework. In section 5.2.2, we cover the tests we run on the persistent tracker and present the results of those tests in section 5.2.3.

5.2.1 The Tracker

If we take our adaptive tracker and allow it to include information gained from before the track was initiated, we get a posterior of the form

$$p(S_t, M|Y_{-\infty:t}). \tag{5.1}$$

With this, all the information in the observations before our initiation at time $t = 0$ updates M. This means that our tracker, when initiated, will have a prior that is relevant to the environment in which it has been deployed. This prior could cover any part of the model: we could develop a prior on the motion model, learning what motions are possible in our environment; we could develop a prior on which modality of observation model is the most effective, switching between the different observation models according to the environment; we could develop a prior on the joint observation-motion model, learning the ways that different sorts of objects move. The quantity of training data becomes a concern for more complex tasks, so we settle on developing a prior for the observation model. We use our past tracking observations to improve our rejection of false targets (i.e. wave clutter). By spending fewer particles on clutter, we will have more particles to follow the objects of interest.

The likelihood $p(Y_t|S_t, M)$ helps us follow salient objects (i.e. contiguous regions of space-time that remain salient longer than our background model would suggest likely). Placing our prior information in this would make our background model more convoluted, and as this is run on our innermost loop we want it to be as lightweight as possible. We choose to store the information in an interest model $p(I|M, Y_{-\infty:t})$, which models the likelihood that an object is a target, given our past observations. In this way, our gradient descent still finds the most likely salient object with a lightweight function, and our interest model is run on the final winner to decide whether it is a target or clutter. This gives us equation 3.46 from the derivation chapter (repeated below)

$$p(I, M, S_t|Y_{-\infty:t}) = p(I|M, Y_{-\infty:t})p(M, S_t|Y_{-\infty:t}). \tag{5.2}$$

Here the second factor is our adaptive tracker, as implemented in section 5.1, and the first factor describes all the learning that occurs before instantiation of the target in question.

We are speaking of the M for a specific track. However, until each track is instantiated and has cropped pixels updating its observation models, all the tracks will have the same information in M. In this way, we can learn a single M_{prior} that is used to instantiate each new track's M, and when the track is over we can use its results to update M_{prior}. To do this, we need each track to be labelled as {Object of interest, Clutter, Neither}[4]. Again we consider our use case to guide our decisions. Most of our clutter is in the form of short-term waves. We could autonomously label any short tracks as waves, and longer tracks as objects of interest. On the other hand, a surveillance system is likely to have a human interacting with it at some level. It would be possible to have the machine present tracks to the human and have them labelled. While the first approach would be interesting to pursue, it would introduce another independent variable into this application. As our intent is to prove SMAE's applicability, we choose to limit our scope. We use labelled data to mark tracks and make this available to the persistent tracker. We feel this is justified for maritime surveillance, as it simulates a human operator.

Once we have labelled data, we still need to decide on how to use it in our prior. This is the quintessential machine learning problem — we have 'things' for which we want to predict a function's output (in this case, a label). We need to make two decisions: in what feature space should we represent our 'things', and what function approximator should we use?

[4]The third category allows us acknowledge that, even though static objects like lighthouses are not targets, they are probably closer to boats than waves. Trying to discriminate against them may lead to inferior results. We will leave these out of the training set.

Our observation model is described by the approximate value for the mode of $\vec{\alpha}$ representing the grid of pixel alpha values. In our literature review, we discussed many of the features used to describe image patches in preparation for function approximation. We chose a selection of these features and others and will test the efficacy of different subsets. We are looking for macroscopic tendencies with which to differentiate (for example: 'few middle values of α', or 'model is one big convex mass'), and so avoid feature sets like Haar or HOG features that are good for describing the fine structure required in frame-to-frame adaptive tracking. These feature sets also have large numbers of dimensions, which may lead to over-fitting our limited data.

Our features can be summarised as follows (where $\vec{\alpha}$ is the 20-by-20 grid of modes in M):

- Quantity and general placement of salience:
  - f1: Mass of salience $= \sum \vec{\alpha}$
  - f2: Effective radius $= \sqrt{\frac{\text{moment of inertia}}{\text{mass of salience}}}$
- Tendency towards α of 0 or 1:
  - f3: Mid-values $= \sum(\vec{\alpha}(1-\vec{\alpha}))$
  - f4: Extreme values $= \sum e^{0.5\left(\frac{\vec{\alpha}-0.5}{0.2}\right)^2}$
- Tendency to contain all salience in a single mass with no holes or other patches:
  - f5: Normalised local consistency favouring gradual changes $= \frac{\sum_{\text{adjacent pixels}} (\alpha_1-\alpha_2)^2}{\text{mass of salience}}$
  - f6: Normalised local consistency favouring sudden changes $= \frac{\sum_{\text{adjacent pixels}} |\alpha_1-\alpha_2|^{\frac{1}{2}}}{\text{mass of salience}}$
  - f7: Morphologically opened mass $= \sum(\vec{\alpha} > 0.5) \circ \text{ones}(5,5)$
  - f8: Morphologically closed mass $= \sum(\vec{\alpha} > 0.5) \bullet \text{ones}(5,5)$
- State variables:
  - f9: Width = perspective-adjusted width of object
  - f10: Height = perspective-adjusted height of object
  - f11: Area = perspective-adjusted area of object
  - f12: X-position = x co-ordinate in frame
  - f13: Y-position = y co-ordinate in frame
  - f14: Frames allocated to view = number of updates this view has
  - f15: Frame index = number of this frame in sequence.

For each feature we consider the feature value and its change due to the most recent update (so we have f1-value and f1-velocity). All features are normalised to be within the same range.

Once we have our labelled data we need to train a classifier. We consider three different classifiers: a K-nearest neighbours classifier (which we will call our KNN persistent tracker), Bayesian inference based on Gaussian approximations (which we will call our Gaussian persistent tracker), and a neural network (which we will call our NN persistent tracker). While it would be possible to do an in-depth investigation into which of the many available classifiers would work best (possibly even building that into M_{prior}), we intend only to prove the applicability of SMAE, and so decide that further optimisation is out of scope.

5.2.2 Description of Tests

To test our persistent tracker, we run our adaptive tracker on every sequence, collecting all the selected features for the models in each tracking output. For each sequence Q, we train persistent trackers using all the sequences that are not from the same original video as Q. We train persistent trackers for all three learning algorithms (KNN, Gaussian, and NN).

We perform a greedy feature selection by testing the trained classifier on the initial tracking samples for Q. This means that the feature selection may be over-fit, however with our constrained data set this was the most reasonable compromise.

Our classifiers then predict $p(I|M, Y_{-\infty:t})$ as follows: KNN takes the fraction of the K (we used $K = 10$) nearest neighbours that are of interest; the Gaussian approximation takes the ratio of likelihoods for the interest and clutter distributions; and the neural network takes its output bounded to the range $[0 : 1]$.

In order to rank our feature sets we consider the number of disordered pairs in the test data, where a pair is disordered if one is a target of interest, the second is clutter, and the classifier rates the second as being more likely to be a target of interest. The difference between the two classifier outputs is counted as a penalty for the classifier. Thus the total penalty for a classifier is

$$\text{Penalty} = \sum_{\text{disordered pairs}} |p(I_1|M_1) - p(I_2|M_2)|. \tag{5.3}$$

Finally, we run the new tracker (which we will call the persistent tracker) on sequence Q. In this way, we simulate information from extended tracking runs with multiple shorter

tracking runs by performing leave-one-out testing on the sequences. We record the same metrics for the persistent trackers as for the adaptive trackers.

5.2.3 Results of Tests

The results for feature selection are summarised in Table 5.9. Because the neural network has a much larger training time, we used a smaller number of starting points for the greedy feature selection (we started from at least[5] 20 random initial sets). The sets chosen are very different, with few features either chosen or ignored in all trackers. Those universally chosen are f4 (discriminates between middle values of α and extreme values), f10 (perspective-independent height of the target), f11 (perspective-independent area of the target), f12 (X-position of the target), and f14 (the number of updates a view has had). The only feature that was rejected by all the trackers was f13 (Y-position of the target). Some of these patterns make sense: figures 5.5, 5.6, 5.7, and 5.8 show that a common failure mode for these types of trackers is blurring, thus f4 which discriminates between certain salience choices (i.e. α close to 0 or 1), and vague values (i.e. α near 0.5) would be helpful. It also makes sense that f14 would be useful: the state of a template is more pertinent if we know how much information has been incorporated into it. More curious is the favouring of X-value and rejection of Y-value, which we would have predicted to be opposite. Targets and waves occur at all horizontal positions, yet clutter seems to have a high occurrence at the lower part of the frame.

It is possible to hypothesise why the feature selection profile is as observed: perhaps the information held in target height is adequate to rule out clutter near the bottom of the frame, and so the Y-position adds no new information. However, this is all speculation. Because we used an ad hoc strategy, it is difficult to tease meaning from results that are not as we expect. As this is an incidental step on the path to our persistent tracker, we are satisfied to use its results in the context of our larger Bayesian framework. It serves as a pertinent illustration of the differences between ad hoc approaches and full Bayesian frameworks. We choose not to look further for meaning in these features, and simply use the best features for each of our persistent trackers. Although the penalty for the Gaussian approximation is much worse than that of the others, we find in the next tests that its actual performance is comparable.

[5] Due to machine failures the exact number is uncertain.

Classifier	% of Search Space Searched	Feature Mask	Penalty
KNN	0.1614	01111 10001 11010	18790.3
Gaussian	0.1628	11010 00011 11010	73773.5744
Neural Network	0.0355	00111 01111 11011	14126.4467

TABLE 5.9: Optimal feature sets as found by greedy feature selection. Each row represents one of the machine learning algorithms considered: K-Nearest Neighbour, Bayesian inference based on a Gaussian approximation, and a neural network trained to classify. For each algorithm we show the percentage of search space that was covered in the gradient descent algorithm (as a percent of the 2^{15} possibilities), the optimal mask found, and the penalty incurred by the optimal mask (as described in 5.2.2).

Figure 5.10 shows the results for the different persistent trackers. We favour the harmonic mean again, presenting the side-by-side comparison of the harmonic and arithmetic means in appendix B. The PR curves' global values may not be impressive[6]; however, we can see that PR curves for all three learning algorithms show an improved overall performance, and that in all but four sequences the adaptive tracker performed worst. We show the MO, MT and SW errors for completeness, but do not find their values particularly significant. The MT and SW errors are lower for the adaptive tracker than for the persistent trackers. The SW errors are caused by a tracker re-establishing contact with a lost target. We see the persistent trackers' higher SW rating in light of their higher recall as a sign that they are re-establishing tracks that the adaptive tracker ignores. We de-emphasise the MT errors for the same reasons given in section 5.1.3.

These are aggregated results and need to be seen in parallel with specific instances. In figures 5.11, 5.12, 5.13, and 5.14 we present the same example frames used in figures 5.5, 5.6, 5.7, and 5.8. We refer to the text in section 5.1.3 for an explanation of the colours used. The small target in sequence 6 is tracked adequately by all four trackers; in these sequences the persistent trackers would be able to improve precision by attaching lower interest probabilities $p(I|M, Y_{-\infty:t})$ to the clutter particles. Similar results are achieved for the small target in sequence 9, except that the KNN-trained tracker lost track of the target before frame 200, and was unable to re-establish the track after the occlusion that occurs between frames 200 and 300. The larger object in sequence 14 is tracked far better by the KNN and NN persistent trackers[7]. Similarly, the static target in sequence 19 was tracked better by both the KNN and the NN persistent trackers.

[6] We note again that our data set is rather challenging (including targets that are very difficult to track), and that the harmonic mean is lower that the arithmetic mean (in this sense it could be thought of as a soft minimum operator).

[7] Frame 100 for the neural network has a rare edge case in the pixel allocation algorithm, where the algorithm cannot assign the pixel allocations and so leaves them blank. The templates show that the target was being tracked by two tracks.

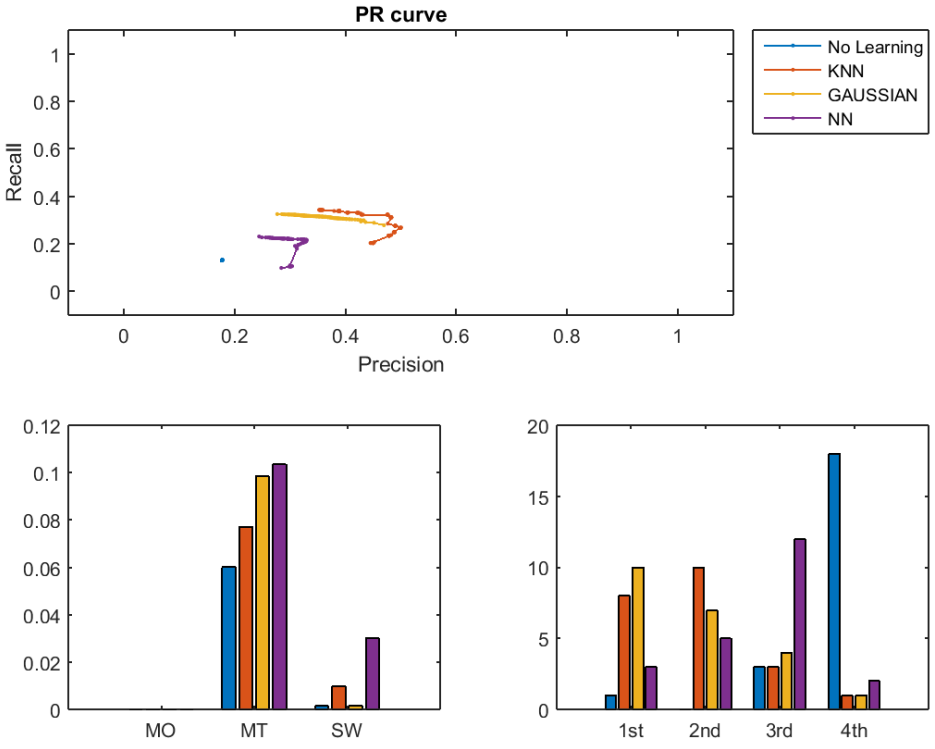
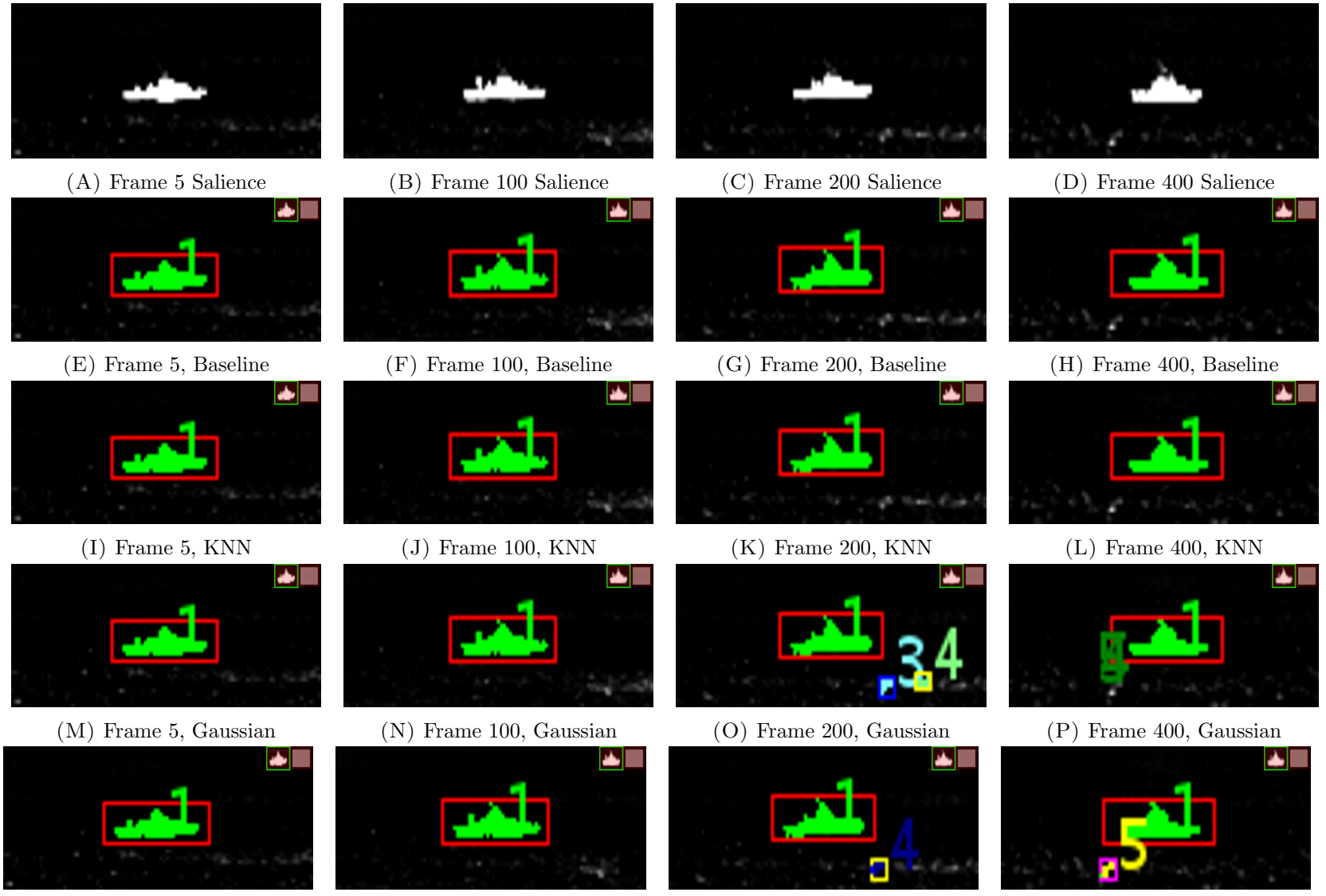


FIGURE 5.10: The top shows PR curves for the three persistent trackers relative to the baseline adaptive tracker. The lower left shows MT, MO, and SW scores for the different trackers. The lower right shows the number of sequences each tracker ranked in each position (as judged by F-score). The MT, MO, and SW errors use the arithmetic mean again, as the harmonic mean of any set with a zero element is zero.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

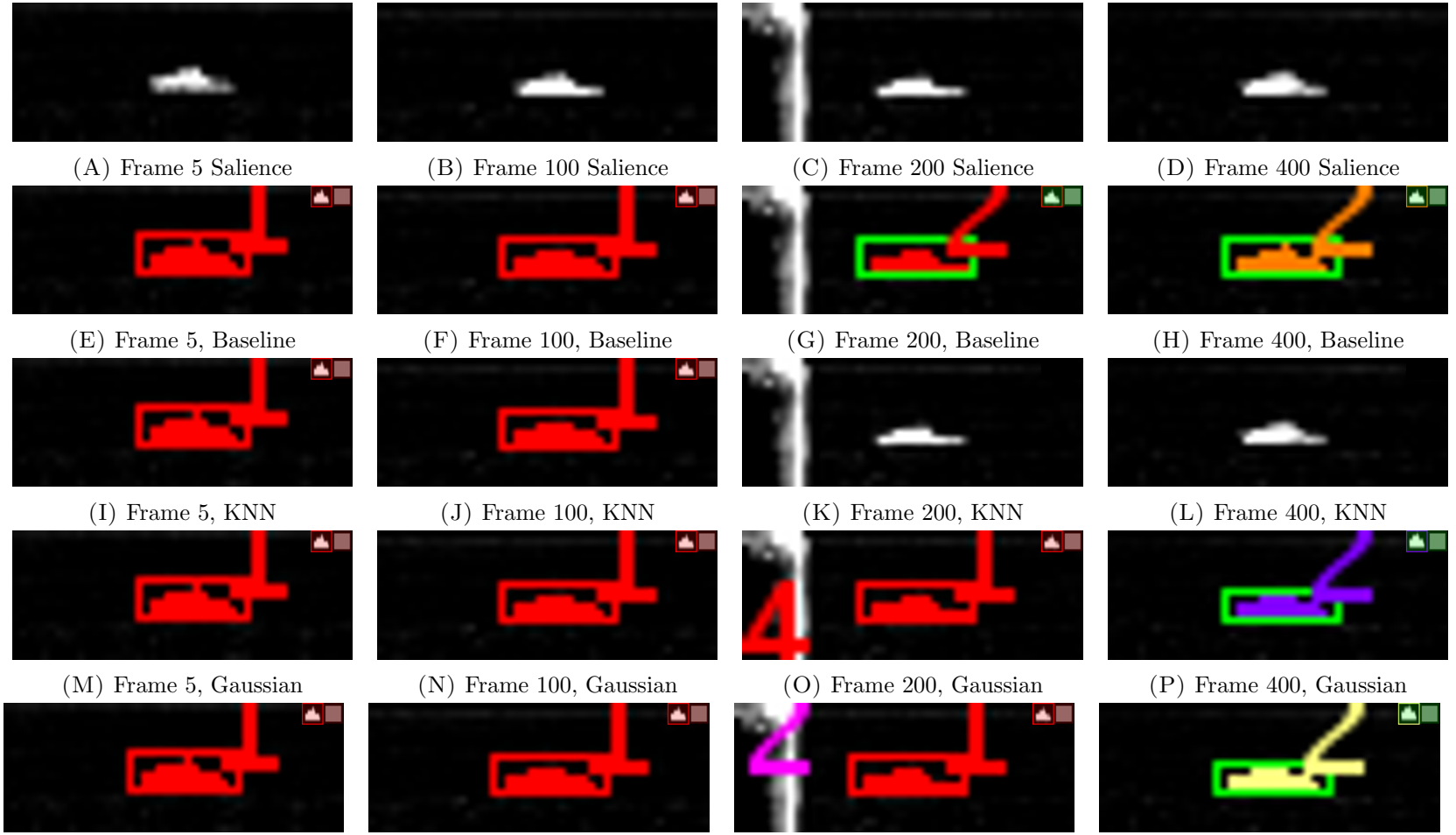
(E) Frame 5, Baseline (F) Frame 100, Baseline (G) Frame 200, Baseline (H) Frame 400, Baseline

(I) Frame 5, KNN (J) Frame 100, KNN (K) Frame 200, KNN (L) Frame 400, KNN

(M) Frame 5, Gaussian (N) Frame 100, Gaussian (O) Frame 200, Gaussian (P) Frame 400, Gaussian

(Q) Frame 5, NN (R) Frame 100, NN (S) Frame 200, NN (T) Frame 400, NN

FIGURE 5.11: Sample templates for a medium-sized target in sequence 6 for persistent trackers with different learning algorithms.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

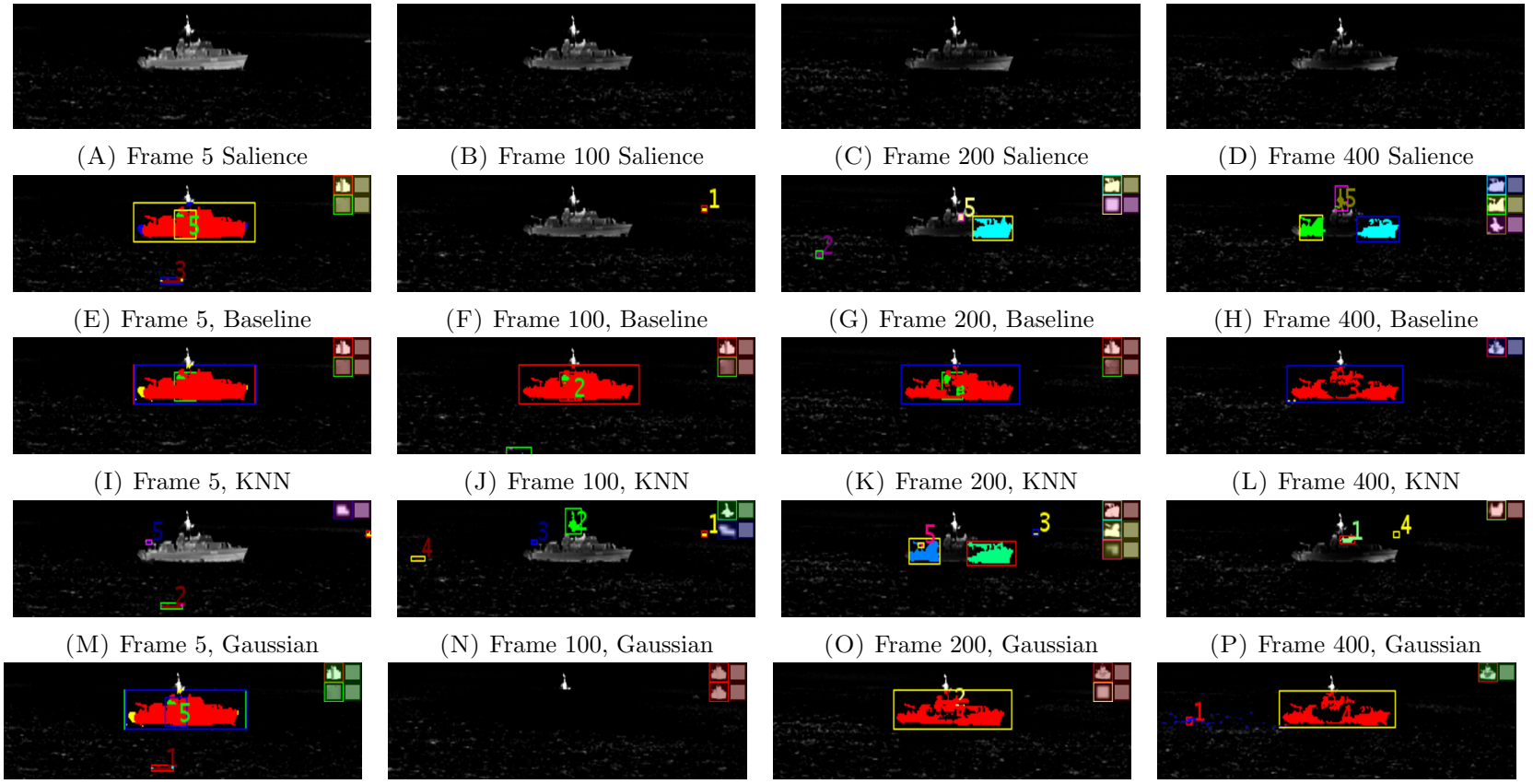
(E) Frame 5, Baseline (F) Frame 100, Baseline (G) Frame 200, Baseline (H) Frame 400, Baseline

(I) Frame 5, KNN (J) Frame 100, KNN (K) Frame 200, KNN (L) Frame 400, KNN

(M) Frame 5, Gaussian (N) Frame 100, Gaussian (O) Frame 200, Gaussian (P) Frame 400, Gaussian

(Q) Frame 5, NN (R) Frame 100, NN (S) Frame 200, NN (T) Frame 400, NN

FIGURE 5.12: Sample templates for a small target in sequence 9 for persistent trackers with different learning algorithms.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

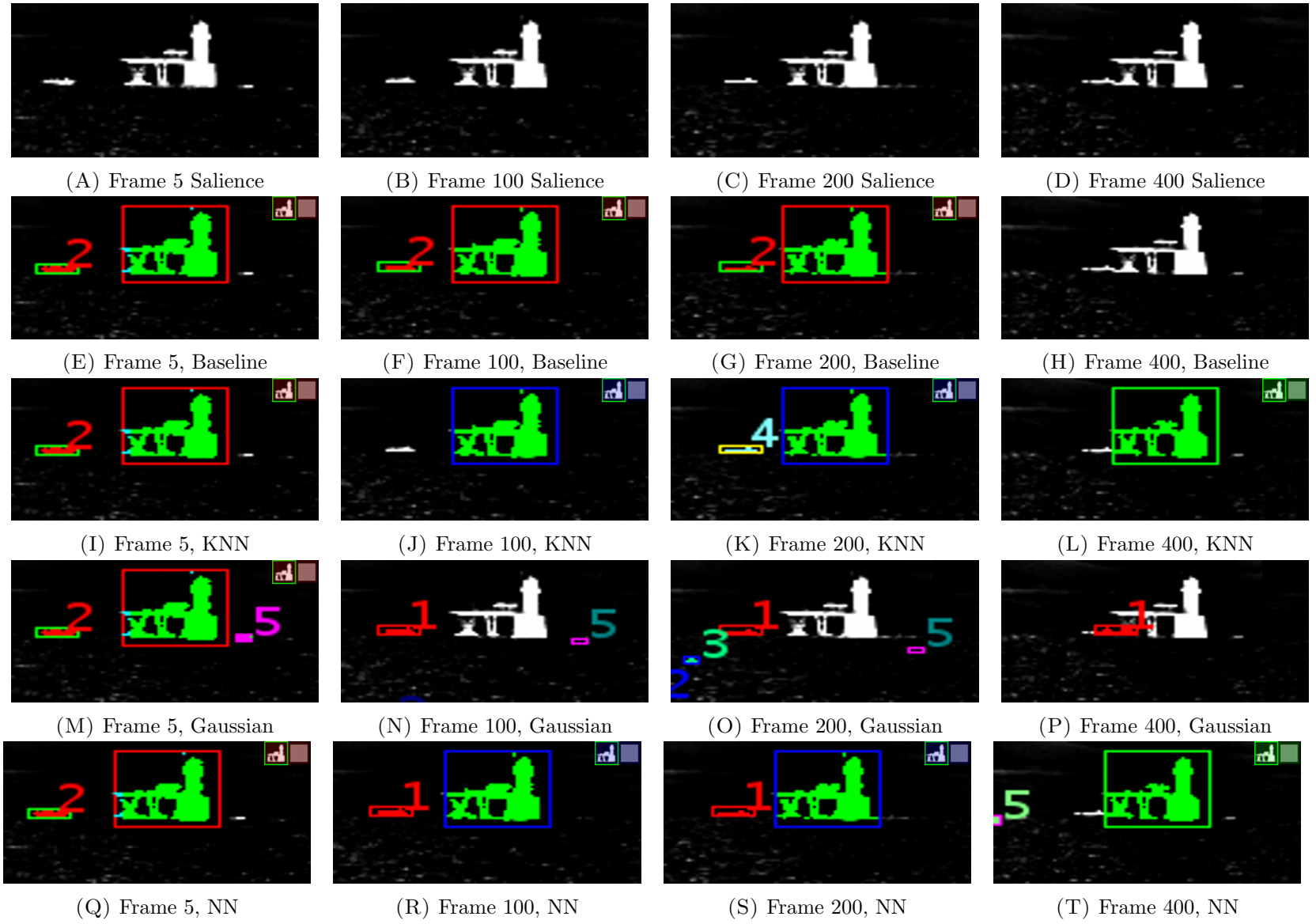
(E) Frame 5, Baseline (F) Frame 100, Baseline (G) Frame 200, Baseline (H) Frame 400, Baseline

(I) Frame 5, KNN (J) Frame 100, KNN (K) Frame 200, KNN (L) Frame 400, KNN

(M) Frame 5, Gaussian (N) Frame 100, Gaussian (O) Frame 200, Gaussian (P) Frame 400, Gaussian

(Q) Frame 5, NN (R) Frame 100, NN (S) Frame 200, NN (T) Frame 400, NN

FIGURE 5.13: Sample templates for a large target in sequence 14 for persistent trackers with different learning algorithms.



(A) Frame 5 Salience (B) Frame 100 Salience (C) Frame 200 Salience (D) Frame 400 Salience

(E) Frame 5, Baseline (F) Frame 100, Baseline (G) Frame 200, Baseline (H) Frame 400, Baseline

(I) Frame 5, KNN (J) Frame 100, KNN (K) Frame 200, KNN (L) Frame 400, KNN

(M) Frame 5, Gaussian (N) Frame 100, Gaussian (O) Frame 200, Gaussian (P) Frame 400, Gaussian

(Q) Frame 5, NN (R) Frame 100, NN (S) Frame 200, NN (T) Frame 400, NN

FIGURE 5.14: Sample templates for a static object in sequence 19 for persistent trackers with different learning algorithms.

We see from both the quantitative and the qualitative results that the persistent tracker as designed by SMAE performed better than the simple adaptive tracker. While this improvement may be moderate, the quantity of learning data was limited and the inclusion of more data would likely increase the improvement. It is worth noting that this provides no information about the saturation results towards which the persistent trackers would tend: it only shows that the model has enabled positive learning for a persistent tracker. A possible extension of this work would be to test these results with many extended sequences (rather than simulating a small subset with leave-one-out testing). Our goal was to test the functionality of SMAE for both adaptive tracking and persistent tracking. In light of this, we are satisfied that we have achieved our objective. The joint (M, S_t) models for template-based tracking are functional, and we are able to include problem-specific learning (as opposed to target-specific learning) into M for our persistent trackers. The algorithm is not real time, but is still processable (I had it running for extended periods of time on a single desktop machine, with improvements to efficiency and more hardware it could be made to run real time). Because the particle filter uses a gradient descent, it is difficult to give a "big Oh notation" complexity class for this algorithm; run time would be proportional to the average gradient descent length, and the number of particles. We conclude this chapter addressing what we assume will be a remaining reservation for the reader.

5.3 SMAE for More Standard Features

We have addressed this tracking problem using raw pixel values as our features, rather than those features that are favoured by current state-of-the-art trackers. We have done this to keep the application of the SMAE as straightforward as possible, including enough detail to show how SMAE deals with challenges, but not so much as to make following the application unnecessarily difficult. Certainly, if we were not using raw-pixel values, the templates shown in figures 5.5, 5.6, 5.7, 5.8, 5.11, 5.12, 5.13, and 5.14 would be much harder to visualise in a clear manner.

However, most current trackers do not use template-based, raw-pixel features. Haar features are common; they are fast to calculate, and very effective in a boosted cascade classifier. HOG features are also common, catching a summary of the gradient of a patch and hence its geometry. There are many different binary descriptor features that are commonly used (SIFT, SURF, ORB, FREAK etc.). These can be used to describe a local patch, but are also commonly used to describe key-points on a larger target. Setting aside tracking using key-points and a parts-based model, we discuss the use

of SMAE with patch features (features that can be seen as a function of the cropped proposed target).

We showed that incorporating the entire frame into Y is equivalent to using a LLR for a considered patch. This makes the SMAE framework easy to extend to patch-based features. One could use these features much as we have used raw pixels. If we considered a target to have a Gaussian observation distribution for certain Haar features, then we can infer a posterior on the parameters of that distribution through our observations. One could model binary descriptors similar to our alpha masks, modelling the probability that a given bit in the descriptor would be positive. In both of these cases it would be harder to visualise templates, and the mathematics becomes involved in resolving which feature is expressed when targets overlap. With these changes it would be possible to extend our framework to non-rigid-body tracking tasks, such as people tracking. The key difference between the SMAE approach and the current approach is holding the observation model in a distribution. Rather than passing training samples to an ad hoc classifier, one should define an observation model for the features such that it can be held inside a probabilistic variable.

As mentioned above, we chose to demonstrate SMAE with raw pixel values as features for the sake of clarity (it is easier to visualise the templates, and the mathematics remains relatively transparent); however, we see no reason why these methods could not be applied with other common features.

All the code used for chapters 4 and 5 is available at `http://www.dip.ee.uct.ac.za/~cbradshaw/PhD_data/`.

Chapter 6

Conclusion

In this chapter we close off the technical side of our work. While there is still content in the next chapter, it is of a more diversionary nature and so we include it as an epilogue. In section 6.1 we present our concluding thoughts, and revisit our most pertinent novel contributions.

6.1 Summary

This document started off by introducing the approach spectrum: a way of thinking about different problem-solving styles, that ranges from the results-focused practical side to the model-accuracy-focused principled end of the spectrum. Bayesian techniques were presented as an example of the principled side of the spectrum. With figure 1.3(B), we illustrated how current Bayesian trackers are primarily practical approaches that use a principled Bayesian component. We hypothesised that it is possible to design a Bayesian adaptive tracker that encompasses the entire adaptive tracking task within a single inference, in a manner that is still tractable.

Through our investigations of the current adaptive Bayesian trackers in sections 1.3 and 2.1, we found that while there are many effective Bayesian trackers, none encompass the holistic framework we envisioned. The adaptive model components of current Bayesian trackers may be probabilistic, in the sense that they are expressed as observation likelihoods, however they are not probabilistic variables themselves. We also found evidence that most current Bayesian trackers see the observation presented to the inference engine as limited to the values within a bounding-box, rather than the values in the entire frame.

In designing our holistic Bayesian adaptive tracker in chapter 3, we addressed both of these concerns. Our tracker includes the observation model (and has the capacity to include the motion model) inside the probabilistic variables tracked by the inference engine. Instead of having an ad hoc process updating the likelihood function, the different likelihood functions compete against one another inside the Bayesian inference. This lead to us naming our framework SMAE, for simultaneous modelling and estimation. We also found that considering the entire frame as the observation is equivalent to using the LLR for the bounding-box as the observation. This particular observation means that the behaviour of feeding a bounding-box into the inference engine can arise from both the holistic Bayesian tracker (figure 1.3(A)) and the current approach (figure 1.3(B))[1].

Our framework also addresses multiple-object adaptive tracking, and how to extend the holistic tracker to a more general task. This leads to our differentiating between an adaptive tracker (one that is instantiated and tracks a single target across frames, while learning its appearance) and a persistent tracker (a long-lived task that encompasses many adaptive tracker tracks, improving through extended deployment). We saw that just as the framework describes how an adaptive tracker can use each frame to update its observation model, it also describes how a persistent tracker can use each adaptive tracker's results to update the prior it carries into future adaptive tracker instantiations.

The derived framework represents the bulk of our contribution, although an untested framework is of no use. In order to demonstrate the value of the developed framework we applied it to the challenging task of maritime surveillance. We did this in chapter 5. However, in order to do so we also needed to survey the relevant literature (section 2.2) and cover some non-framework specific topics (chapter 4). Chapter 4 is noteworthy in that, although it is not the focus of our work[2], it still contains novel contributions (as listed below). The work in chapter 5 shows how SMAE can be applied to real-world challenges, and how it can be made to accommodate reasonable approximations to result in a tractable principled framework. Section 5.1.1 highlights the links between the derivation in chapter 3 and the trackers in chapter 5, describing how the framework applies to our use case.

Chapter 5's instantiation of SMAE uses raw pixel values as the input features. Section 5.3 addresses why this design decision was made, and how the framework could

[1]This leads to a typically Bayesian situation: the same evidence (using a bounding-box for the observations of the inference engine) supports both hypotheses (the approach modelled by figure 1.3(A) and the approach modelled by figure 1.3(B)). This casts doubt on our claim that current papers do not model the entire frame as the observation. We acknowledge this, yet we are unable to find further evidence of the holistic Bayesian framework in current works.

[2]It is merely covering ground to set up chapter 5, which in turn exists to verify the work in chapter 3.

be extended to more common tracking features. Investigating the scalability of performance with respect to the number of tracked targets would be another interesting path for future work.

6.2 Contributions

Before moving to the less concrete applications of SMAE in the next chapter, we close off by listing our contributions in the light of our finished work. Our key contributions are that we:

- Illustrate the inappropriateness of the BRE to adaptive tracking. Although our discussion of the time-invariance of the model in the BRE is brief, it is pertinent in light of the prevalent use of the BRE for adaptive trackers. Without a framework that acknowledges time variant information's effect on the model, it is impossible to approach adaptive tracking in a principled manner.

- Formulate a holistic Bayesian adaptive tracking framework that incorporates model estimation into the PDF. SMAE, as a framework, addresses the adaptive tracking task in a principled manner we found missing in the literature. It includes the model uncertainty inside the Bayesian framework in a manner more consistent with Bayesian reasoning than the common approach of inserting an ad hoc learning algorithm into the observation model.

- Draw attention to the difference between an observation model in the form of a probability distribution on the outputs, and one that is held in a distribution as a probabilistic variable itself. It is easy to get confused when a distribution is itself a probabilistic variable. Distinguishing between a probabilistic observation model (one that gives different probabilities for different possible observations) such as is common for generative trackers, and a distribution of observation models (in which the model is uncertain, and exists in a PDF) was an important part of constructing a principled framework for dealing with this uncertainty.

- Emphasise and prove that considering the whole frame as observation (which is the most principled way to approach tracking) is equivalent to considering the contrast between the foreground observation and the background observation models for a bounding- box. It may seem that we proved that trackers can continue doing what they are already doing, however this would miss the importance in the difference between labelling the bounding-box as Y, and labelling the entire frame as Y. It is obvious that the entire frame should be considered Y; the machine observes every

pixel. Most classifiers want features from the bounding-box only, so it is convenient to consider only the bounding-box as Y. That we can do what is convenient in a principled manner is important. Also, considering the values of the observation Y changing depending on what hypothesis is being considered is an abuse of the HMM structure.

- Develop a MTT particle filter that handles the multiple hypothesis in a novel manner and avoids the holding of joint solutions. By normalising across possible subsets of particles, we maintain the structure of a single PDF over the (M, S_t) space, without experiencing the domination by one target or the coalescence associated with using a single PDF.

- Create a salience filter for maritime surveillance that is an improvement on the current state of the art. Our use of a limited neural network trained to predict pixel value from pixel co-ordinates combines the benefits of salience filters that look for salience by time, and those that look for salience by frame. The results are an improvement on the current standards which, while sufficient for calm seas, are not adequate for the conditions common in our data set.

- Highlight an appropriate measure of data fusion for maritime salience filters. While the use of a Bayesian data fusion for multiple probabilistic salience filters is not an inspired step, it is lacking in the literature. Thus we point it out in the hope that it will find more use.

- Develop a persistent maritime tracker that improves at wave rejection in a principled manner. While the primary focus of this work is the mathematical framework, our persistent tracker (that is, one that improves over time learning from each tracking run for subsequent tracks) does provide a contribution to its field: a principled manner for learning to reject waves from the tracker's responses.

Chapter 7

Epilogue

While developing the theory required for this work, we discovered several 'softer' applications of SMAE and faced a difficult choice: we could either leave out these insights, which have played a large part in our development of SMAE, or attempt to shoe-horn them into a technical topic. As this is a Doctorate of Philosophy, we feel that is is not out of place to explore the philosophic aspects of our work, and so present these 'meta' applications of SMAE in an epilogue separate to the main body of work. They show the broader applicability of the technical framework, and in turn are grounded in the preceding concrete application.

We start by applying SMAE to interpersonal interactions in section 7.1, and explore an implication of this in section 7.2. We apply SMAE to social rituals in general in section 7.3, and close off in section 7.4 by looking at this document in light of this discussion.

7.1 Introduction to Interpersonal SMAE

It is neither a novel nor an inspired step to suggest that Bayesian inference should be useful in understanding the way people communicate. It is a truism that people seldom say what they mean[1], that we as listeners use all the evidence from the statement, choice of words, tone, body language, topics avoided, etc. to infer the intended underlying message is a given. This is the straightforward application of Bayesian methods: X is the message the sender is trying to send, Y is all the evidence they present, thus infer $p(X|Y)$[2].

[1] A insightful discussion of this is found in chapter 11 of Robert McKee's book on screenwriting [53].

[2] This implies that the listener needs an adequate set of hypotheses $\hat{X}$ such that at least one is sufficiently close to the underlying X, but more on that later.

Many non-transactional interactions go beyond simply passing a message between two parties. When we relate to people, it is not that we perfectly understand the person, and are uncertain of the message. The idea is laughable. No, we are trying to simultaneously model the person (motives, trustworthiness, unconscious tells, communication strategies, relationships with common third-parties, etc.) while we are processing and responding to the message being sent. This is exactly the (M,S_t) separation we have been dealing with in the previous chapters. The distribution is a joint PDF across 'the possible person I could be talking to'[3] and 'the possible message he or she could be sending'.

Under different considered person-models M, different messages S become more likely. Our response as listeners is often more dependent on M than it is on S. Consider a person asking you to do a favour for them in a socially appropriate way (e.g. talking parallel to the issue, implying a need, and then leaving a space for you to offer help). If your model of them (M) is that of a friendly colleague who struggles to ask for help, you are likely to help them. On the other hand, if your model is that of a manipulating laggard, you will decline (possibly in a socially acceptable way by changing the subject without noticing the implicit request).

The listener is often more focused on localising this distribution in M than in S. This is especially true of social interactions; we appear to spend more energy deciding who a person is, and managing who they believe we are, than we do on the actual information being sent back and forth. This is a complex 2-player game, and the messages S often contain content about M, making the system more convoluted. When one party sends a message 'I value X,' the other party may (correctly or not) infer, 'It is a problem if you do not value X', 'I am offended by Y', or 'I trust you enough for self-revelation'. All of these unintentional messages have implications on which M are supported and which are rejected.

There is an innate problem in this setup. The space of messages (S) on its own is of larger dimensionality than the space of evidences Y (many hypothesised messages can be constructed for the same evidence). When you include the dimensions associated with M, you get an extremely ill-defined inference problem. Any evidence Y supports infinitely many (M,S) hypotheses, with an equally infinite range in responses. As demonstrated in the 'favour request' example above, the result of the inference leads to diverging interactions. In situations like this, the prior is what must differentiate the hypotheses. So it is the prior distribution on hypotheses that predict the same Y that will determine which dominate, and which are perpetually hidden.

[3] That is: Who is the person really? What are their goals? What attributes do they have? What do I know about them?

Thus an intrinsic problem with communication is that the sender is confined to supporting a member of the set of prior hypotheses the receiver is considering. If we trivialise a diner commenting on a meal to a host: the set of evidences Y are {Diner: 'it was good', Diner: 'it was not good'}. The host's hypothesis set is {It was bad and the diner told me so; It was bad but the diner lied because he did not want to hurt my feelings; It was good but the diner lied because he wanted to hurt my feelings; It was good and the diner told me so}. Consider the case where the host's priors on this set are {0.4, 0.4, 0.1, 0.1}. The latter two hypotheses will be rejected because any evidence that would support them also provides support for another hypothesis with a larger prior. In such a case, it is impossible for the diner to select evidence from Y that will compliment the host. He is confined to communicating the messages in S that are in the host's accessible hypothesis set: {It was bad and the diner told me so; It was bad but the diner lied because he did not want to hurt my feelings}. One might suggest the diner keep quiet. However, 0 is a number; saying nothing is a possible evidence set. We have increased Y's cardinality to 3, but the underlying problem persists.

This effect occurs in a straight Bayesian application without M, but is even more devastating with the added dimensionality that M provides. If the above host's hypotheses included elements dealing with the diner's M, it may include aspects like: his value for the host, his stress levels from his day, his possible tendency to use compliments to manipulate, etc. Because there are only two possible Y's this extensive hypothesis set will be cut down to two: one for a positive comment, one for a negative. The diner cannot reinforce a hypothesis outside this set.

The receiver is justified in appealing to Newton's flaming laser sword (NFLS) [54], which we will paraphrase as, "it is not worth debating the relative worth of hypotheses that predict the same observations[4]." Indeed, from a Bayesian standpoint this is self-evident. If $p(Y|X_1) = p(Y|X_2), \forall Y$, then what observation could support one over the other in inference? This does leave the sender at the mercy of the receiver's prior.

We desire a short-hand for this constraint on the sender (where he can only provide support for a limited subset of the (M,S) space due to the many-to-one, hypothesis-to-evidence, message-to-communications relationship). We will call it the inaccessible message problem (IMP).

As mentioned above, the IMP is applicable if only inferring S, but is even more extensive when M is included. This is not surprising when considering how convoluted a space M is. A listener may have a prior indicating links between certain pronunciation, choice of

[4]Similar to how Von Neumann (or Dirac/Eckart/Schrödinger, depending on how you read the histories) silenced the debate over Heisenberg's matrix mechanics and Schrödinger wave equation by proving that they are equivalent on every observation [55].

clothes, or distance between eyes, and desirable or undesirable attributes. These priors, will occur in different M locations for different people and may cut in either direction. This makes non-transactional interactions a mathematical minefield for any optimising intelligent agent; and yet people manage.

7.2 Rational Disagreement Based on the Same Evidence

We follow a brief tangent related to the IMP, before returning to our discussion on SMAE for interpersonal interactions.

There are many conspiracy theories that are seen as irrational (climate change denialists, flat Earth-ers, anti-vaxxers, etc.). I do not want to descend into politics or conspiracy theories, but instead draw attention to section 5.3 of Jaynes' book [2], "Converging and diverging views". He shows the same statement heard by two rational entities can diverge their opinions, based on the evidence not being 'X', but 'A says X'. This is in line with our emphasis on modelling the joint PDF across both the message S, and the model of the person M. Our point here is not for or against any specific cause, we merely note and draw attention to the fact that IMP also acts on populations. When we take the same evidence (which is almost always second-hand, so of the form 'A says X') that we believe is conclusive because of our priors, and consider it in light of a doubter's priors, we may find them acting rationally. After all, NFLS suggests it is meaningless to discuss the difference between the hypotheses 'it is good to drink water' and 'there is a sufficiently resourced, sufficiently informed interest group that wants me to believe water is good to drink'. Appealing to the likes of Occam's razor is no proof; we have said that a Bayesian approach is the only way a rational entity can assign values to uncertain statements[5]. If Bayesian reasoning says the observation has equal likelihood from both hypotheses, and this pattern holds across $Y_{-\infty:t}$, then it comes down to the prior. At this point, there are no observations left to support or deny any hypotheses, or to justify Occam's razor itself, which we may accept only after Bayesian inference accepts it.

This is a maddening tangent to descend into: if one tries hard enough, one can construct a set of priors in the (M,S) space that makes anyone's behaviour seem rational. By NFLS, we may consider them as that rational entity. If this is so, why should one strive to act rationally oneself? Having seen the madness at the end of this tunnel, we retreat and follow another path of thought, as even the above discussion may unhinge the author's grasp on reality.

[5] Conditions are given by Jaynes [2].

One might ask where the original priors used by an individual come from. The same way our tracker updated the starting model for each future track based on the results of a current track, we update the priors we will use on interactions with future acquaintances based on our current acquaintances. But there must be a prior used for the first interaction. We believe that this is one of the reasons that we as a species have an obsession with stories. Even though one may never have personally experienced a gunfight, a collapsing building, terminal diagnoses, or political intrigue, one has priors associated with the different events that could occur in these situations. These priors can only have originated from stories one has heard of such events (e.g. movies, fairy-tales, parables, etc.). These stories are not optimised over accuracy of probabilities[6], and should cast serious doubt on one's priors. Yet we move on with them anyway.

One might hope that selective pressures would equip us with priors that are useful. Stories definitely fit the model of memes that can spread and mutate and respond to selective pressures. Unfortunately, the selective pressure is not towards survival of an individual, but dominance of the species. Consider the prior 'when I take risks, I survive against the odds'. This is clearly a dangerous prior for an individual to have. Yet it would help a species spread, defend its vulnerable young, cross dangerous oceans, and compete in tight ecological niches. Risk-taking is dangerous for an individual, but a few survivors can repopulate a species on the other side of danger. Thus the prior is advantageous for a species (hence should be selected for), yet negative for an individual (as this thinking will lead towards danger). A proper analysis would need to be far more involved, yet in light of this we find the proliferation of stories about success against the odds deeply concerning.

7.3 SMAE for Social Rituals

We back off from these possibly disturbing thoughts, and pick up where we left off with interpersonal SMAE. There are many human rituals that revolve around people trying to get an accurate model of one other, while possibly influencing the other's model of themselves. We will focus on that of 'the interview'.

Before we discuss 'the interview' we need to touch on the Turing test. Most current presentations of the Turing test have a human invigilator asking questions to the entity being tested. If the human believes the entity is a human, then the entity passes the test. The test as it is originally presented is slightly different. In his original paper [56], Turing presents two versions of the same three-person game. The game is as follows: an invigilator (C) gets to ask questions to two contestants (A and B). C is trying to

[6] How many royal flushes are seen in movies?

determine which contestant is B, and both contestants are trying to be selected as B[7]. This three-player game describes an interview. The invigilator C is interacting with an unknown candidate, trying to determine whether he is of the desired class (B), or if he is of the undesired class (A), pretending to be B. The invigilator prompts for discriminating evidence, and the candidate tries to give evidence that leads the invigilator to conclude positively. This applies to obvious interviews like job applications or funding proposals, but also to less formal interviews, like meeting new social groups, or dates.

It is in C and B's interest to develop a language through which they can identify one another, which A does not know. Yet for any instantiation of 'the interview', there is a wealth of advice in self-help books/blogs or equivalent media on how to 'look the part'. There is a subtle yet perceivable change in the tone of advice from 'show the interviewer you fit the spot' (i.e. if you are B, let it show through) to 'get that dream job' (i.e. whether you are A or B, convince C that you are B). In this information arms race (between {C; B} and A), the inferences being done are generally on M, not S.

Of course, this analysis is equally applicable to the interview we are currently conducting.

7.4 SMAE for this Document

You, the reader, are currently inspecting all the evidence the author has produced to decide whether he is worthy of a PhD[8]. While the purpose of a PhD is nominally to make a significant contribution to the state of the art, the reality is that every field is so large, with many universities producing PhD graduates in each of them, that it is impossible for an examiner to know for certain whether a contribution is in fact novel. It would appear the job of the examiner has changed from verifying that the work is a substantial, worthwhile, novel contribution in the field, to verifying that the author is of a standard that contributes substantial, worthwhile, novel work. If we think in the (M ,S) plane, the work that has been completed and presented is S. Previously, the goal of the marker would be to verify that this work is of PhD quality. Unfortunately, the novelty clause in the description, when combined with sheer quantity of PhDs, means

[7] While the second version of this game (in which B is human and A is a machine) is similar to current presentations, the paper starts off suggesting that if the invigilator does no better with {A,B}={Machine, Man} than with {A,B}={Man, Woman}, the machine would have passed. This is a subtly different test that draws attention to the test not being of intelligence, but of deception. If a machine can successfully deceive us, we are no longer qualified to judge its intelligence.

[8] I assume here that the reader is an examiner. While I hope that this document's content will be of worth to further readers, if we are all honest the primary purpose of this document is to prove I am worthy of the title Doctor of Philosophy. If its primary goal were contributions to the field, I would segment this work into several digestible minimum-publishable-units and submit them to journals. If my assumption is wrong, and a casual reader has made it this far into the philosophising chapter of a technical work, I offer my deepest apologies and genuinely surprised gratitude.

that no marker can verify this while still conducting their own research. Thus the marker ends up marking the researcher (that is, M), not the research. I suspect that this is a distasteful sentiment to the reader. Certainly, if I were a marker, I would want to mark the work, not the candidate (it is a far better-posed task); however, this is the academic climate in which we find ourselves.

Unfortunately, the IMP applies here too. There will be correlations in the reader's prior that will change from reader to reader, and these will dramatically influence whether they read this as the work of class A or class B: perhaps an informal tone is linked to sub-par rigour in the reader's experience, making the author's discussional style distasteful; perhaps long compound sentences and a tendency to draw from an extended word-set has co-occurred with authors hiding a lack of content behind flourishes of language. No tone is safe, and for a set of markers, almost any tone will have negative indications to at least one. Thus any prospective author needs to have enough evidence throughout the document to prove he or she is in fact of PhD quality, even given a marker in negative sentiment override.

To make matters worse, a strong indicator against class B is a document that is too long. Filling a document with as much evidence of competence as possible is counter-productive. Indeed, what is left out is just as informative as what is put in. Spending limited space explaining fundamentals is acceptable for a Masters thesis. However, in a PhD it displays a disconnection between the author and the field[9]. A surplus of unexplained sample images shows an author inserting diagrams for the sake of diagrams, rather than valuing each page as the proxy for reader interest that it is. Leaving sample images out looks like a lack of rigour — we complained about exactly this regarding maritime literature — and explaining the relevance of each image leads to marker fatigue. Thus the pitch at which the work is presented is one of the most important features of a dissertation. If there is too much detail, the author displays his ignorance for all to see; if there is too little, a lack of rigour and substance becomes his downfall.

This problem of pitching is exacerbated by the novelty criterion for a PhD. The author needs to state why the presented work is a substantial ground-breaking contribution that is lacking from the current work. Yet he needs to do so in a way that neither belittles nor offends his readers, the very people whose work he is at some level calling deficient. This seems an intractable problem, yet there are countless existence-proofs that solutions exist.

This interview is made harder in that I cannot read you as I type. I hope that you have seen the novel contributions in chapters 3 and 4, the technical strength of SMAE

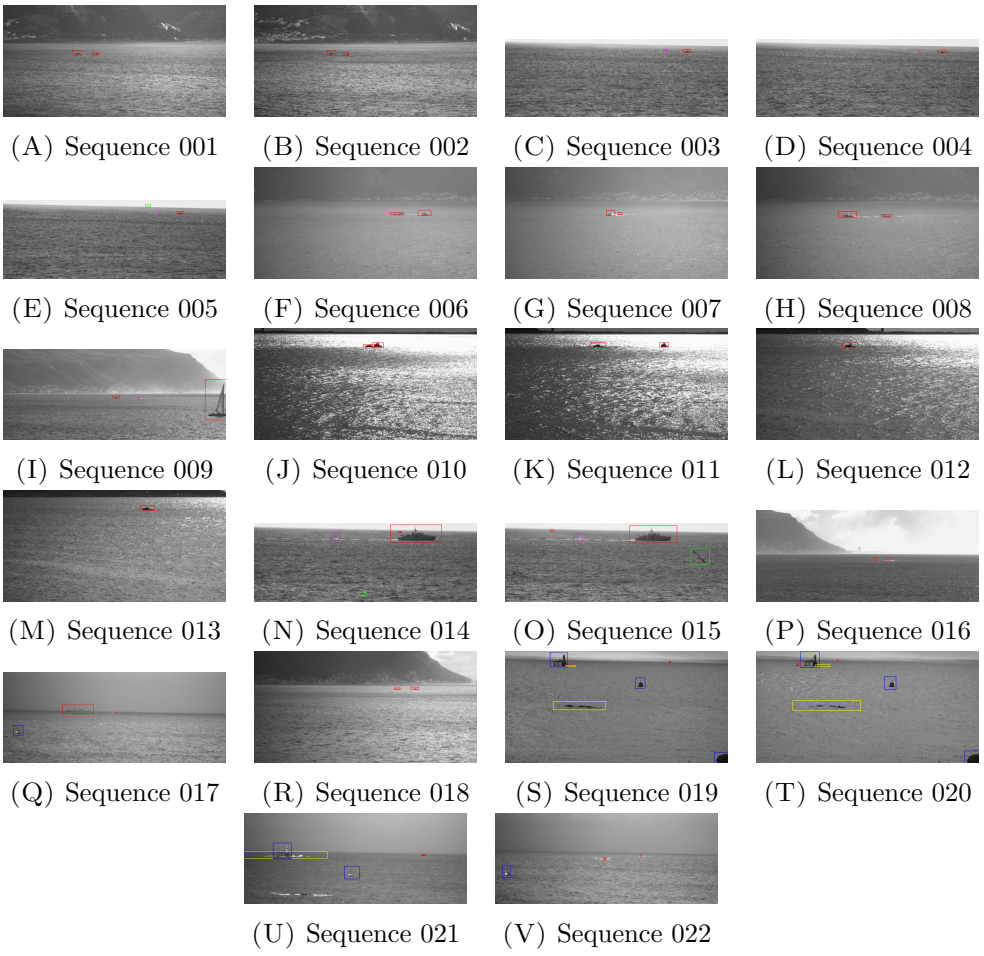
[9]This was particularly challenging for our work where the contribution is close to tools that are common in our field yet are not used to their full strength.

in chapter 5, my competence as a researcher in chapter 1, 2 and 6, and the power of SMAE to unearth the boggy mire that is human interaction in this chapter. I finish off on a lighter note as I know how painful long documents can be, and hope that this finishing chapter has been an interesting diversion after the involved mathematics and technical details of chapters 3 through 5. I have in this closing paragraph broken the rule against the singular first person, but it is in order that my closing sentiment can be more personal. A field such as ours can only exist because there are giants on whose shoulders newcomers can stand, and because those giants choose to undertake the tedious tasks of supervision and marking theses. I offer my heart-felt gratitude to you, the marker, for the effort you have gone through to get a fair assessment of my work, and of me as an academic.

Appendix A

Salience Results

This appendix contains a larger subset of images created by the salience filters. More are available at `http://www.dip.ee.uct.ac.za/~cbradshaw/PhD_data/`.



(A) Sequence 001 (B) Sequence 002 (C) Sequence 003 (D) Sequence 004

(E) Sequence 005 (F) Sequence 006 (G) Sequence 007 (H) Sequence 008

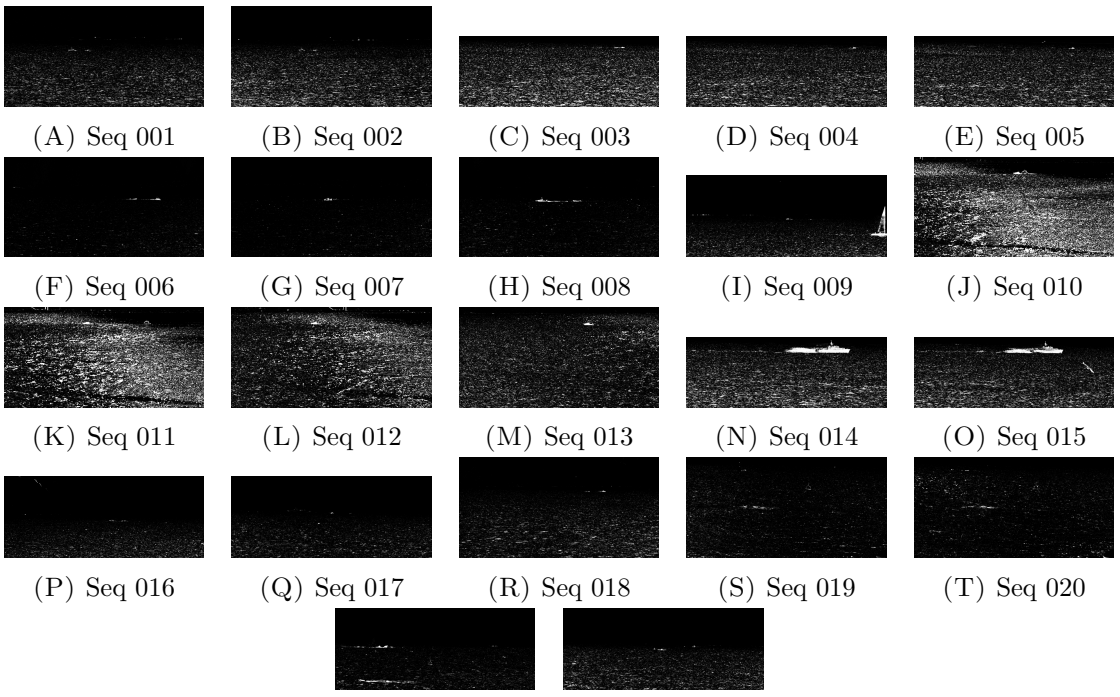
(I) Sequence 009 (J) Sequence 010 (K) Sequence 011 (L) Sequence 012

(M) Sequence 013 (N) Sequence 014 (O) Sequence 015 (P) Sequence 016

(Q) Sequence 017 (R) Sequence 018 (S) Sequence 019 (T) Sequence 020

(U) Sequence 021 (V) Sequence 022

FIGURE 1.1: Sample images for the input sequences.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

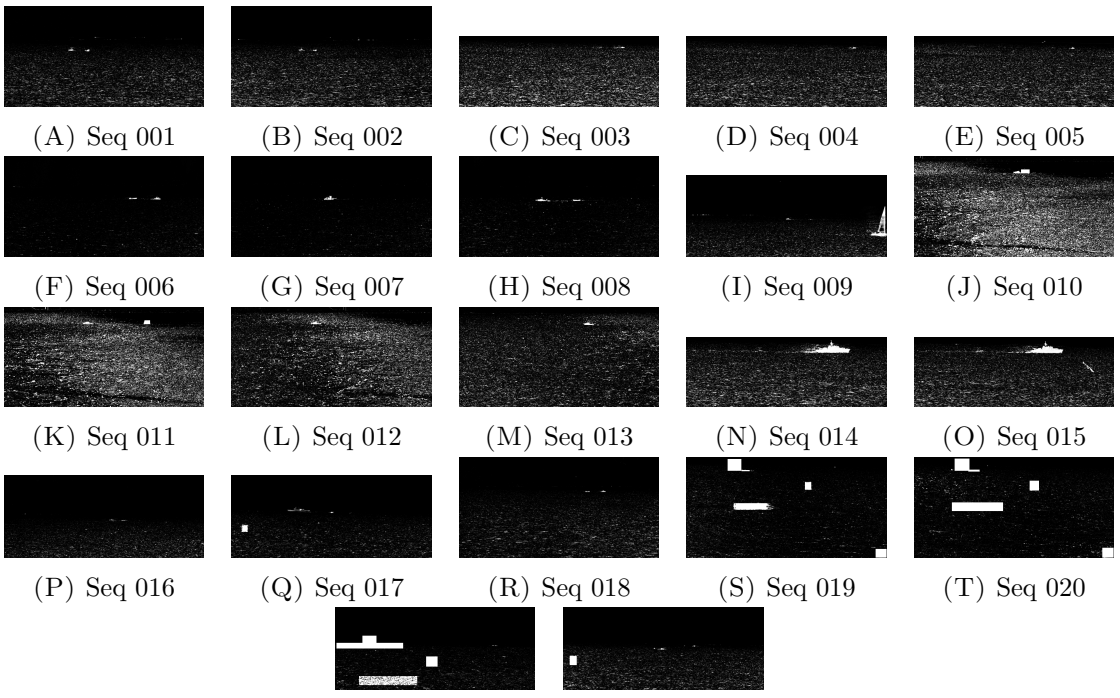
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.2: Sample salience results for 1a-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

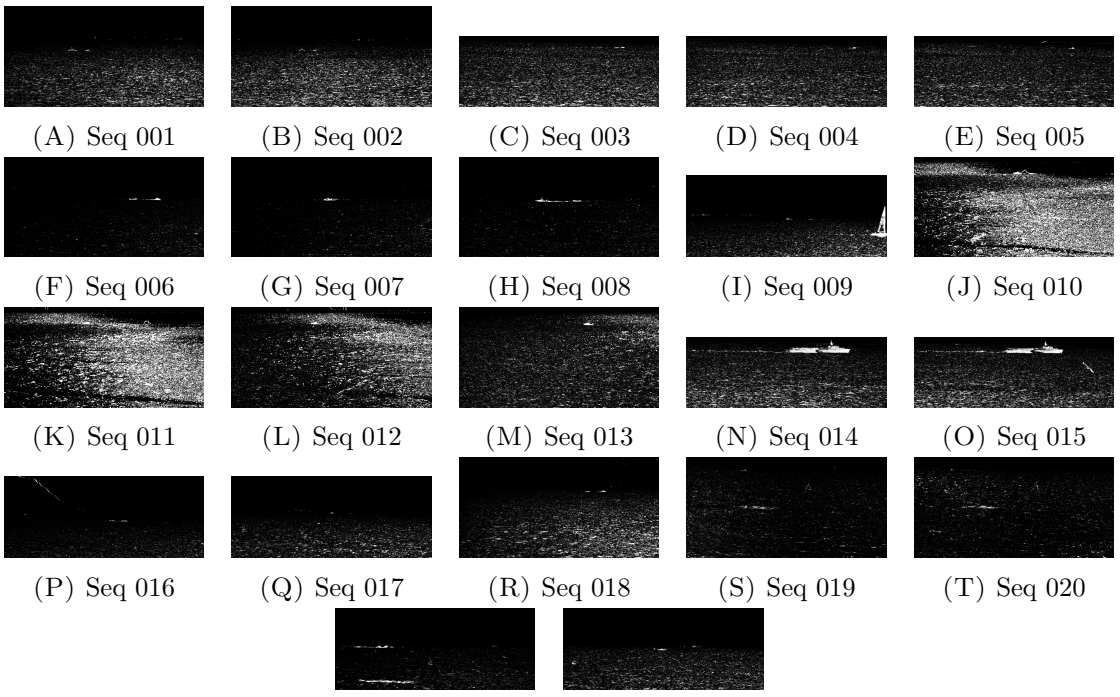
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.3: Sample salience results for 1a-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

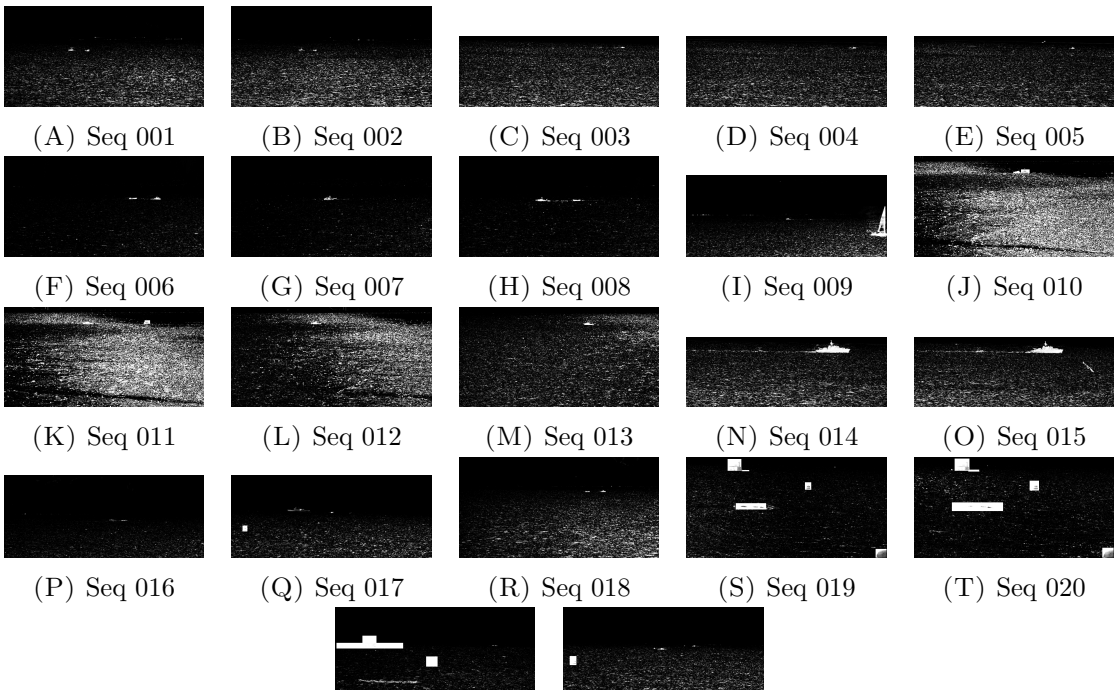
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.4: Sample salience results for 1b-1



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

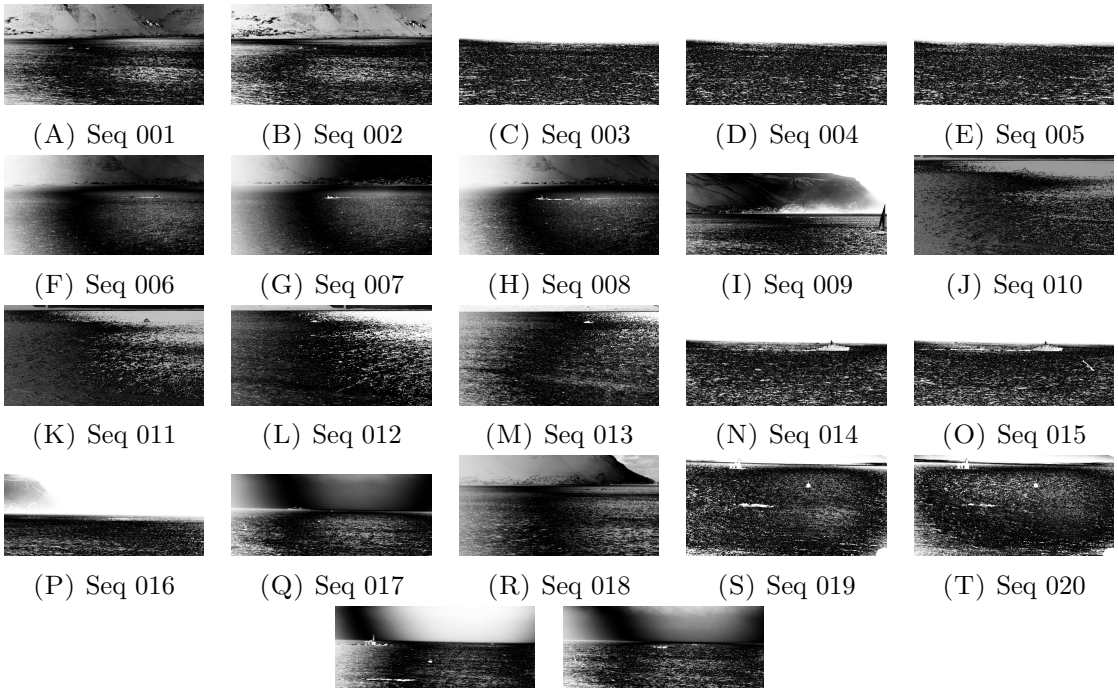
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.5: Sample salience results for 1b-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

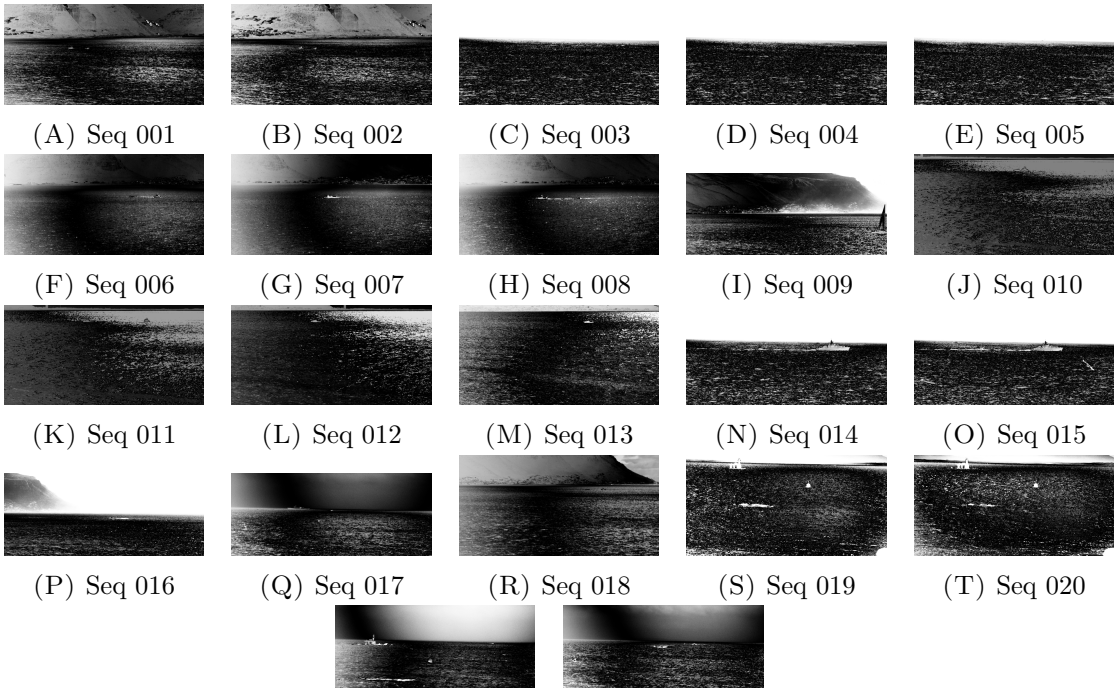
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.6: Sample salience results for 2a-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

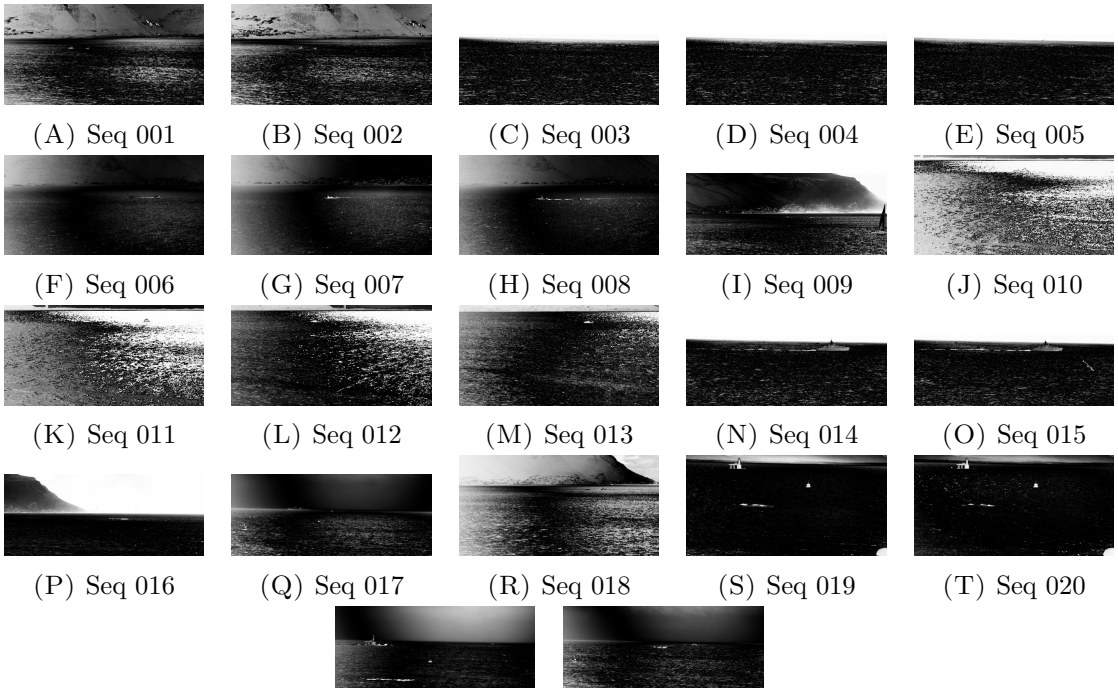
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.7: Sample salience results for 2a-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

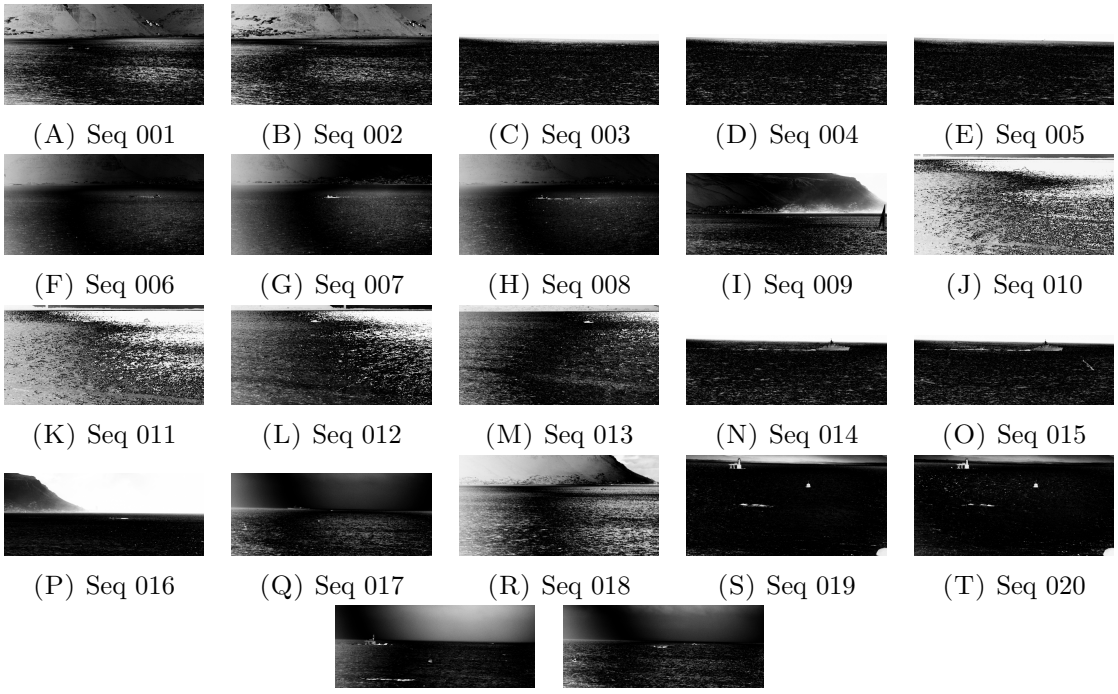
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.8: Sample salience results for 2b-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

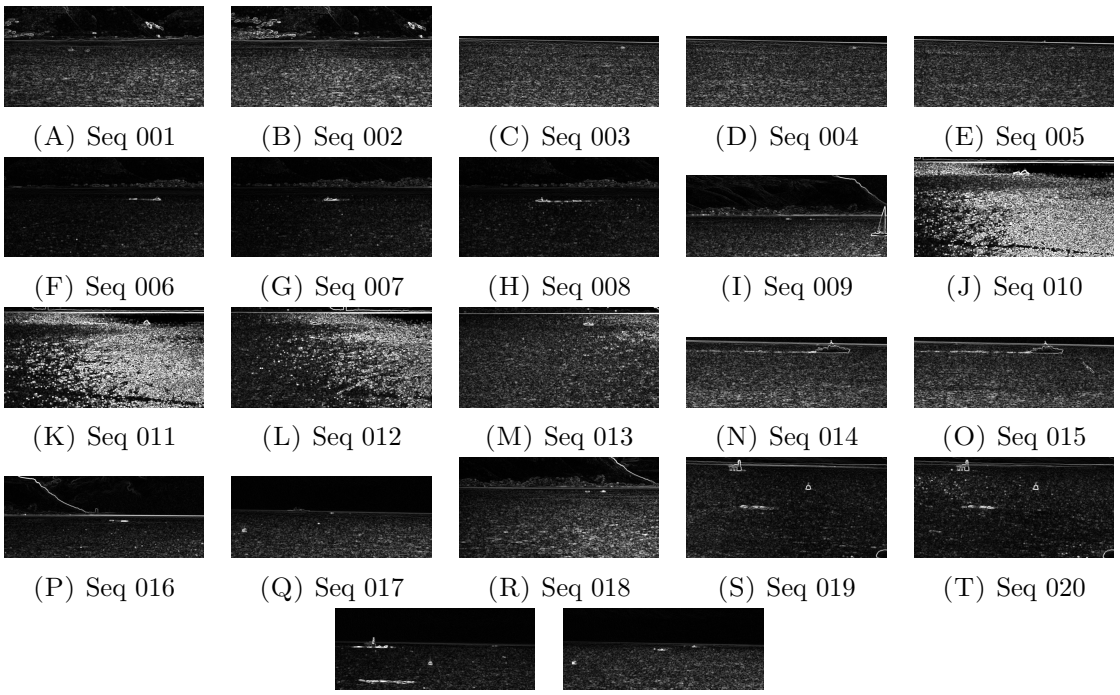
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.9: Sample salience results for 2b-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

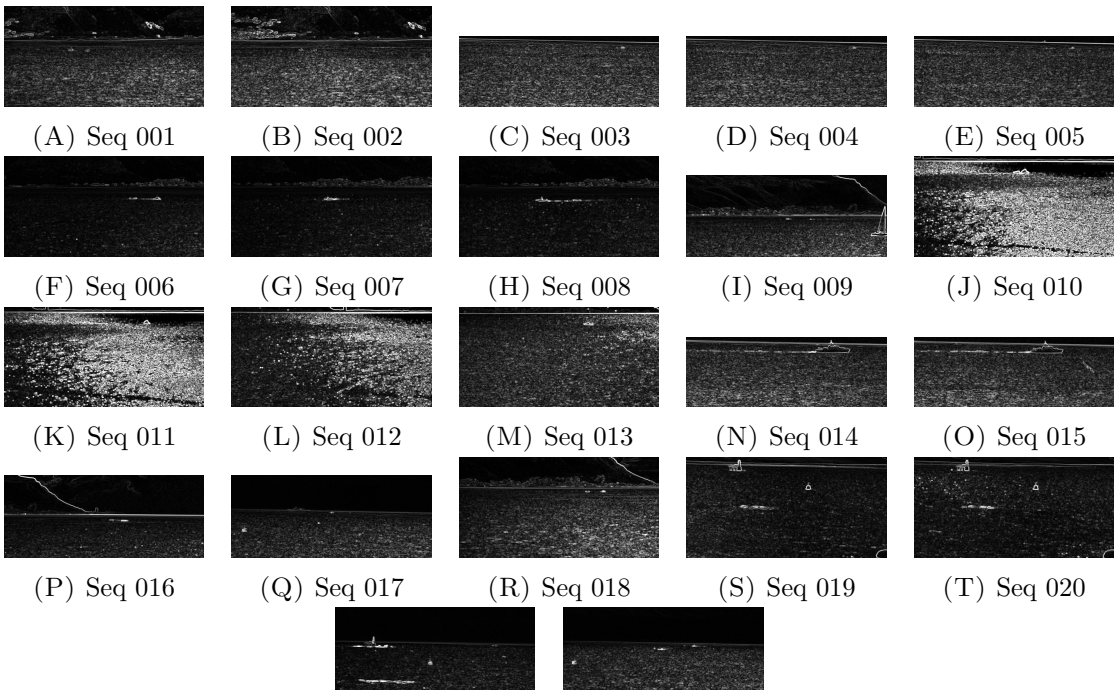
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.10: Sample salience results for 3-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

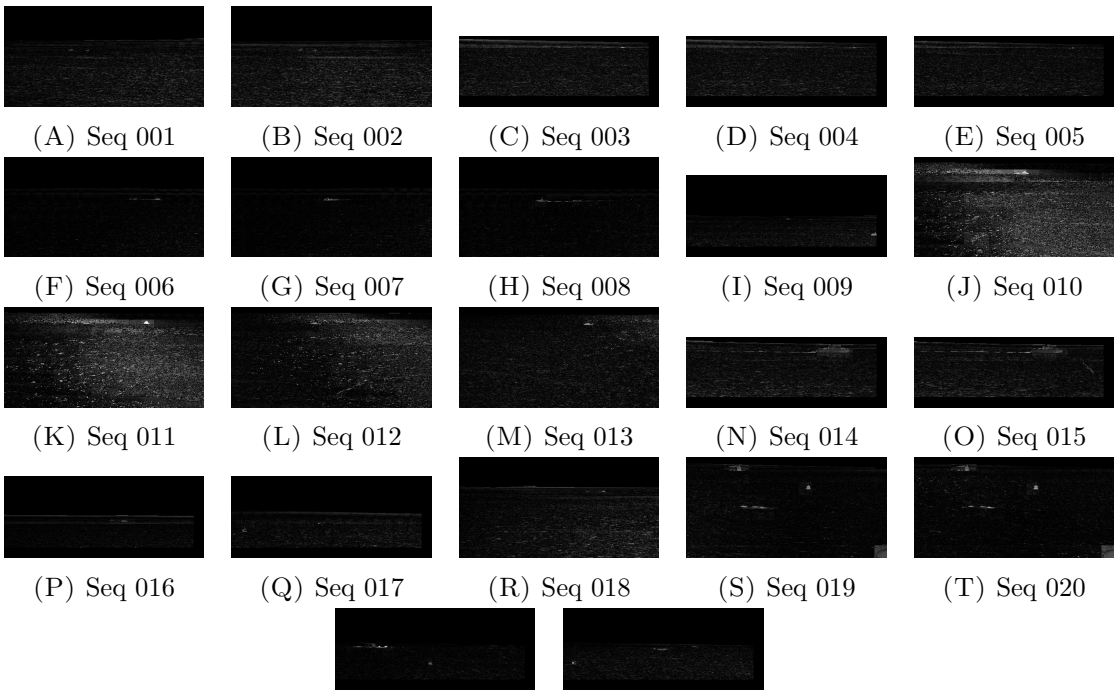
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.11: Sample salience results for 3-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

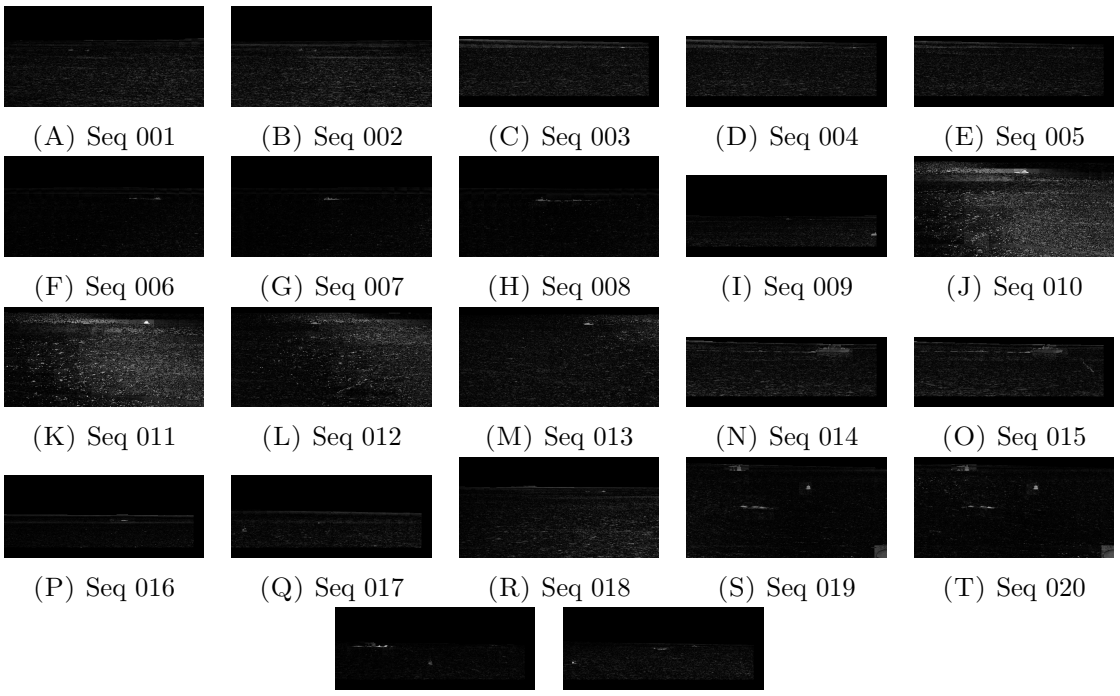
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.12: Sample salience results for 4-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

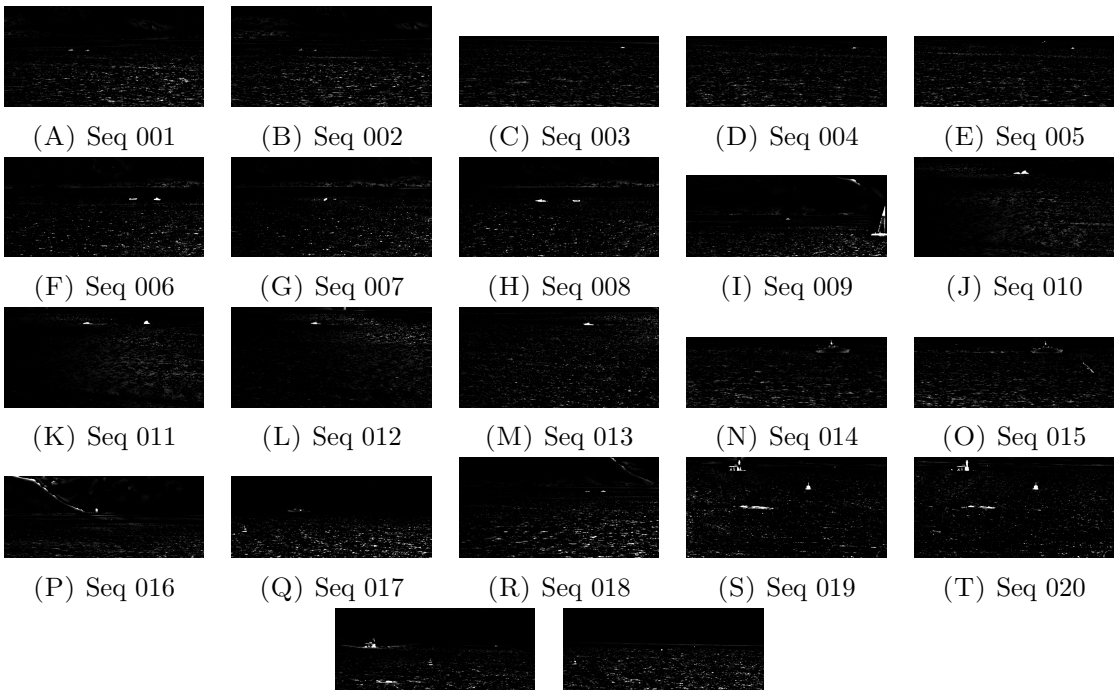
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.13: Sample salience results for 4-2.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

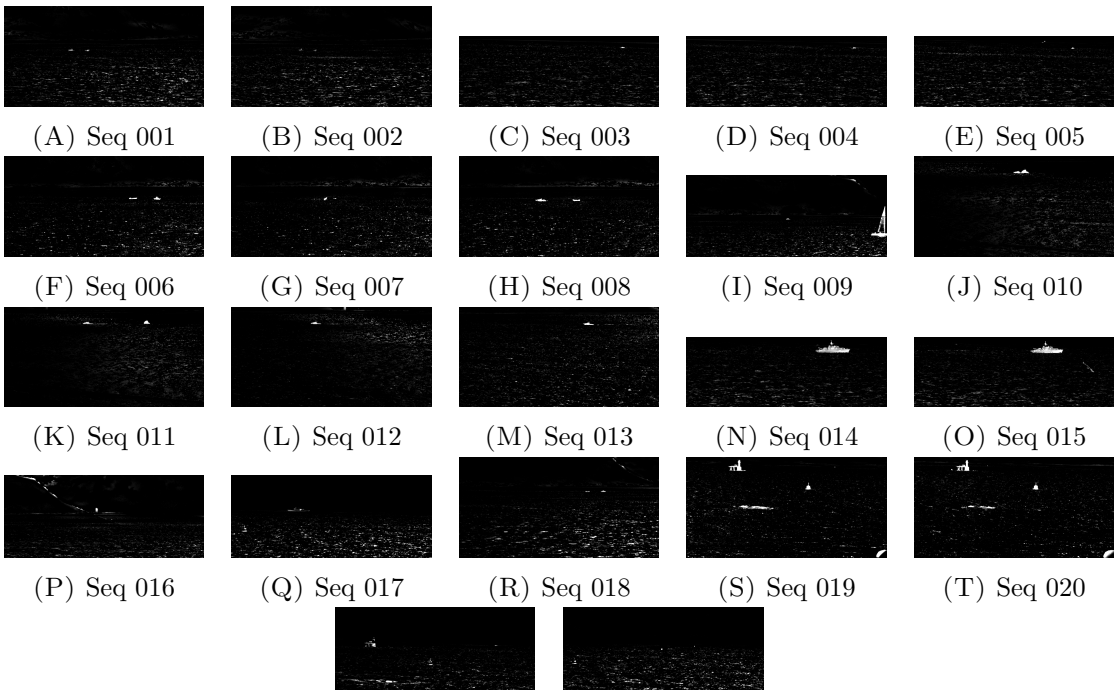
(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.14: Sample salience results for 5-1.



(A) Seq 001 (B) Seq 002 (C) Seq 003 (D) Seq 004 (E) Seq 005

(F) Seq 006 (G) Seq 007 (H) Seq 008 (I) Seq 009 (J) Seq 010

(K) Seq 011 (L) Seq 012 (M) Seq 013 (N) Seq 014 (O) Seq 015

(P) Seq 016 (Q) Seq 017 (R) Seq 018 (S) Seq 019 (T) Seq 020

(U) Seq 021 (V) Seq 022

FIGURE 1.15: Sample salience results for 5-2.

Appendix B

Arithmetic and Harmonic Mean Results

This appendix contains a comparison of the results from chapter 5 as calculated with the geometric and harmonic means. Under the arithmetic mean the improvements are not as pronounced, however we feel the switch to the harmonic is justified due to its results being closer to our intuitive response to the qualitative results. Additionally the harmonic mean favours consistent results over results with a large spread.

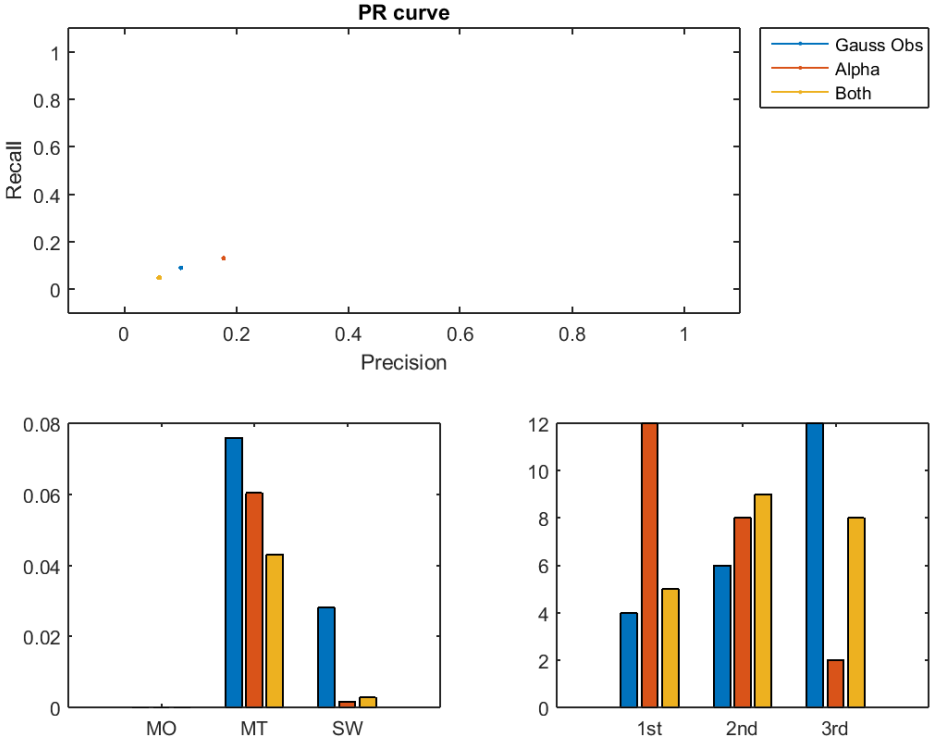


FIGURE 2.1: Adaptive tracker results as calculated with the harmonic mean.

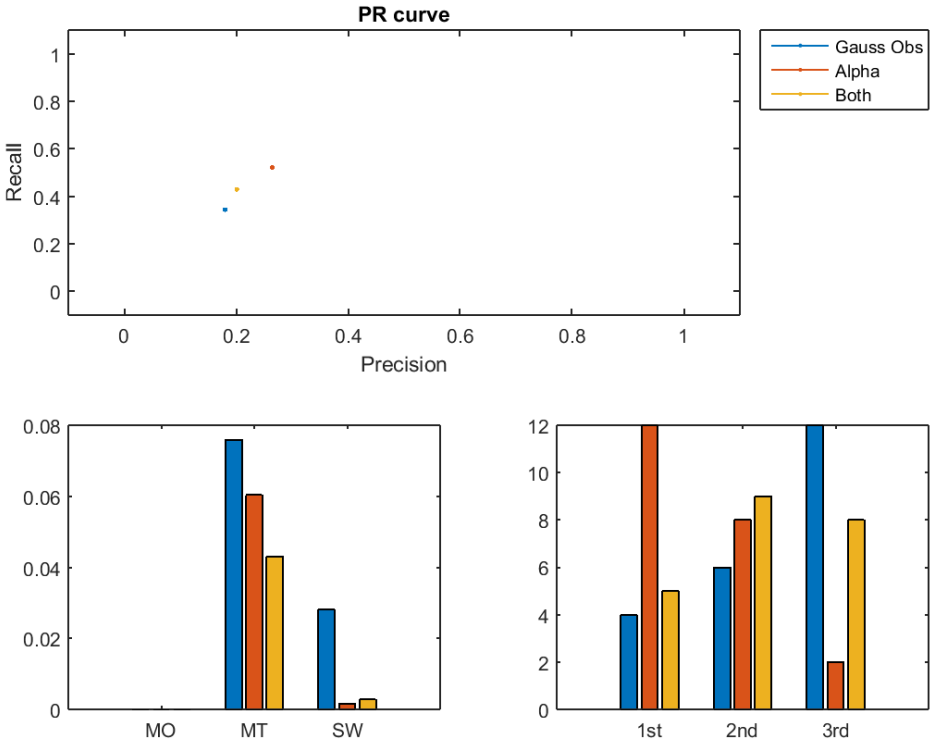


FIGURE 2.2: Adaptive tracker results as calculated with the arithmetic mean.

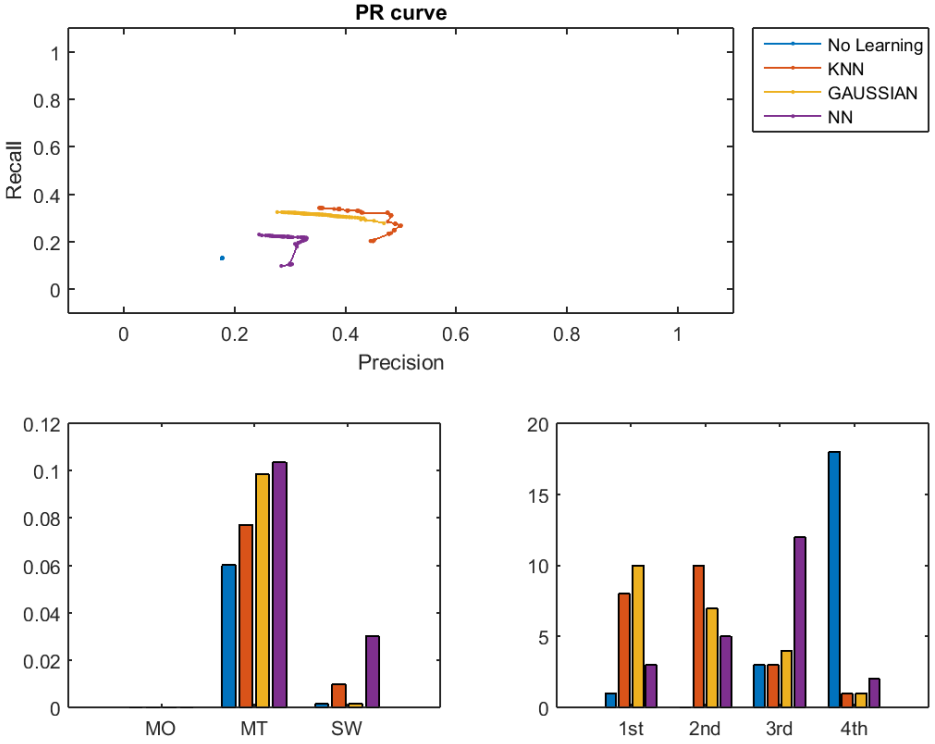


FIGURE 2.3: Persistent tracker results as calculated with the harmonic mean.

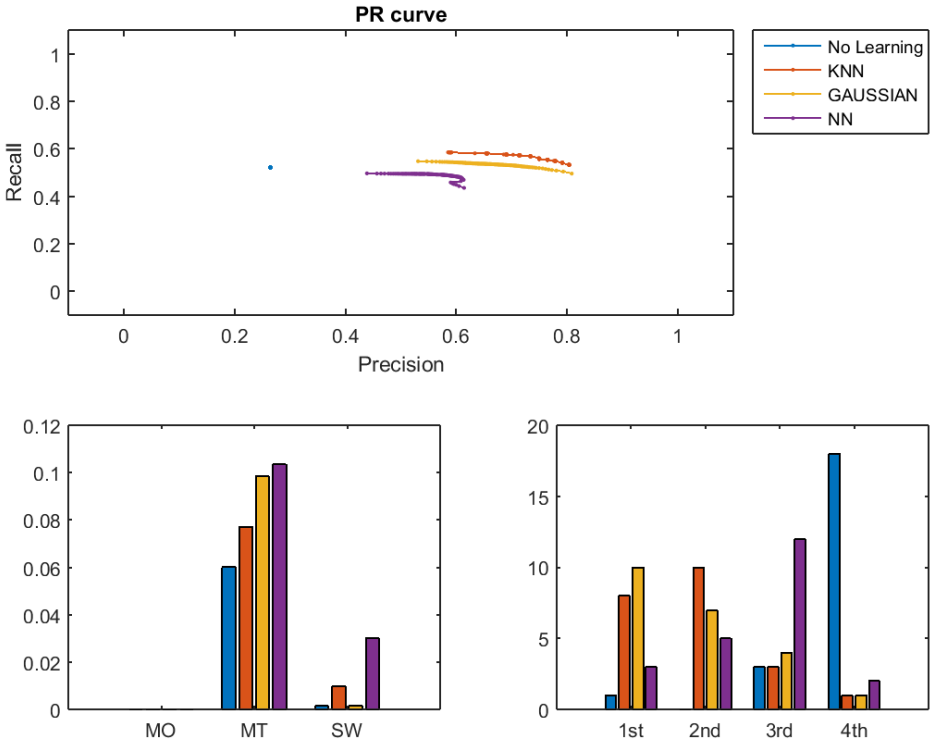


FIGURE 2.4: Persistent tracker results as calculated with the arithmetic mean.

Bibliography

[1] Asheer K. Bachoo, Francois Le Roux, and Fred Nicolls. An optical tracker for the maritime environment. In *SPIE Defense, Security, and Sensing*, pages 80501G–80501G. International Society for Optics and Photonics, 2011.

[2] Edwin T. Jaynes. *Probability theory: the logic of science*. Washington University St. Louis, MO, 1996.

[3] Patrick Pérez, Carine Hue, Jaco Vermaak, and Michel Gangnet. Color-based probabilistic tracking. In *European Conference on Computer Vision*, pages 661–675. Springer, 2002.

[4] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via structured multi-task sparse learning. *International journal of computer vision*, 101(2):367–383, 2013.

[5] Xue Mei and Haibin Ling. Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11):2259–2272, 2011.

[6] Junseok Kwon and Kyoung Mu Lee. Visual tracking decomposition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1269–1276. IEEE, 2010.

[7] Boris Babenko, Ming-Hsuan Yang, and Serge Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990. IEEE, 2009.

[8] Alessio Dore, Mauricio Soto, and Carlo S Regazzoni. Bayesian tracking for video analytics. *IEEE Signal processing magazine*, 27(5):46–55, 2010.

[9] Donald Reid. An algorithm for tracking multiple targets. *IEEE transactions on Automatic Control*, 24(6):843–854, 1979.

[10] Domenico Bloisi and Luca Iocchi. Argos—a video surveillance system for boat traffic monitoring in venice. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(07):1477–1502, 2009.

[11] Thomas E. Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Multi-target tracking using joint probabilistic data association. In *Decision and Control including the Symposium on Adaptive Processes, 1980 19th IEEE Conference on*, pages 807–812. IEEE, 1980.

[12] Zia Khan, Tucker Balch, and Frank Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *European Conference on Computer Vision*, pages 279–290. Springer, 2004.

[13] Rob Hess and Alan Fern. Discriminatively trained particle filters for complex multi-object tracking. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 240–247. IEEE, 2009.

[14] Dieter Koller, Joseph Weber, and Jitendra Malik. Robust multiple car tracking with occlusion reasoning. In *European Conference on Computer Vision*, pages 189–196. Springer, 1994.

[15] Lawrence D. Stone, Thomas L Corwin, and Carl A Barlow. Bayesian Multiple Target Tracking, Artech House. *Inc., Norwood, MA*, 1999.

[16] Domenico D. Bloisi, Fabio Previtali, Andrea Pennisi, Daniele Nardi, and Michele Fiorini. Enhancing automatic maritime surveillance systems with visual information. *IEEE Transactions on Intelligent Transportation Systems*, 2016.

[17] S. S. Blackman, R. J. Dempster, M. T. Busch, and R. F. Popoli. IMM/MHT solution to radar benchmark tracking problem. *IEEE Transactions on Aerospace and Electronic Systems*, 35(2):730–738, 1999.

[18] Henk A. P. Blom and Edwin A. Bloem. Interacting multiple model joint probabilistic data association avoiding track coalescence. In *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, volume 3, pages 3408–3415. IEEE, 2002.

[19] Kevin Smith, Daniel Gatica-Perez, Jean-Marc Odobez, and Sileye Ba. Evaluating multi-object tracking. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)-Workshops*, pages 36–36. IEEE, 2005.

[20] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246309, 2008.

[21] Rodrigo Da Silva Moreira, Nelson Francisco Favilla Ebecken, Alexandre Soares Alves, Frédéric Livernet, and Aline Campillo-Navetti. A survey on video detection and tracking of maritime vessels. *International Journal of Research and Reviews in Applied Sciences*, 20(1):37, 2014.

[22] Kalyan Moy Gupta, David W. Aha, Ralph Hartley, and Philip G. Moore. Adaptive maritime video surveillance. In *SPIE Defense, Security, and Sensing*, pages 734609–734609. International Society for Optics and Photonics, 2009.

[23] Günter Saur, Stéphane Estable, Karin Zielinski, Stefan Knabe, Michael Teutsch, and Matthias Gabel. Detection and classification of man-made offshore objects in TerraSAR-X and RapidEye imagery: Selected results of the DeMarine-DEKO project. In *OCEANS 2011 IEEE-Spain*, pages 1–10. IEEE, 2011.

[24] Michael T. Wolf, Christopher Assad, Yoshiaki Kuwata, Andrew Howard, Hrand Aghazarian, David Zhu, Thomas Lu, Ashitey Trebi-Ollennu, and Terry Huntsberger. 360-degree visual detection and target tracking on an autonomous surface vehicle. *Journal of Field Robotics*, 27(6):819–833, 2010.

[25] Sergiy Fefilatyev, Dmitry Goldgof, Matthew Shreve, and Chad Lembke. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, 54:1–12, 2012.

[26] Wu-Chih Hu, Ching-Yu Yang, and Deng-Yuan Huang. Robust real-time ship detection and tracking for visual surveillance of cage aquaculture. *Journal of Visual Communication and Image Representation*, 22(6):543–556, 2011.

[27] Haiying Liu, Omar Javed, Geoff Taylor, Xiaochun Cao, and Niels Haering. Omnidirectional surveillance for unmanned water vehicles. In *The Eighth International Workshop on Visual Surveillance-VS2008*, 2008.

[28] Gellert Mattyus. Near real-time automatic vessel detection on optical satellite images. In *ISPRS Hannover Workshop*, pages 233–237. ISPRS Archives, 2013.

[29] Charles Bibby and Ian Reid. Visual tracking at sea. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1841–1846. IEEE, 2005.

[30] Da-Jinn Wang, Wen-Sheng Chen, Thou-Ho Chen, and Tsong-Yi Chen. The study on ship-flow analysis and counting system in a specific sea-area based on video processing. In *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on*, pages 655–658. IEEE, 2008.

[31] Daniel Socek, Dubravko Culibrk, Oge Marques, Hari Kalva, and Borko Furht. A hybrid color-based foreground object detection method for automated marine surveillance. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 340–347. Springer, 2005.

[32] Zygmunt L. Szpak and Jules R. Tapamo. Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set. *Expert systems with applications*, 38(6):6669–6680, 2011.

[33] Duncan Frost and Jules-Raymond Tapamo. Detection and tracking of moving objects in a maritime environment using level set with shape priors. *EURASIP Journal on Image and Video Processing*, 2013(1):1–16, 2013.

[34] F. Robert-Inacio, A. Raybaud, and É. Clément. Multispectral target detection and tracking for seaport video surveillance. *Proceedings of the IVS Image and Vision Computing New Zealand*, pages 169–174, 2007.

[35] Wenbing Tao, Hai Jin, and Jin Liu. Unified mean shift segmentation and graph region merging algorithm for infrared ship target segmentation. *Optical Engineering*, 46(12):127002–127002, 2007.

[36] A. A. Smith and M. K. Teal. Identification and tracking of maritime objects in near-infrared image sequences for collision avoidance. In *Image Processing And Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, volume 1, pages 250–254. IET, 1999.

[37] Mohammad Moinul Islam, Mohammed Nazrul Islam, K Vijayan Asari, and Mohammad A Karim. Anomaly based vessel detection in visible and infrared images. In *IS&T/SPIE Electronic Imaging*, pages 72510B–72510B. International Society for Optics and Photonics, 2009.

[38] M Uma Selvi and S Suresh Kumar. A novel approach for ship recognition using shape and texture. *International Journal of Advanced Information Technology*, 1 (5):23, 2011.

[39] Nuno Pires, Jonathan Guinet, and Elodie Dusch. ASV: an innovative automatic system for maritime surveillance. *Navigation*, 58(232):1–20, 2010.

[40] Hai Wei, Hieu Nguyen, Prakash Ramu, Chaitanya Raju, Xiaoqing Liu, and Jacob Yadegar. Automated intelligent video surveillance system for ships. In *SPIE Defense, Security, and Sensing*, pages 73061N–73061N. International Society for Optics and Photonics, 2009.

[41] Michael Teutsch and Wolfgang Krüger. Classification of small boats in infrared images for maritime surveillance. In *2010 International WaterSide Security Conference*, pages 1–7. IEEE, 2010.

[42] Jason G. Sanderson, Martin Kenneth Teal, and Tim J Ellis. Characterisation of a complex maritime scene using fourier space analysis to identify small craft. In *Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, volume 2, pages 803–807. IET, 1999.

[43] Mikel D. Rodriguez Sullivan and Mubarak Shah. Visual surveillance in maritime port facilities. In *SPIE Defense and Security Symposium*, pages 697811–697811. International Society for Optics and Photonics, 2008.

[44] Ikhlef Bechar, Frederic Bouchara, Thibault Lelore, Vincente Guis, and Michel Grimaldi. Uncertainty fusion based object recognition and tracking in maritime scenes using spatiotemporal active contours. In *Computer Vision Theory and Applications (VISAPP), 2014 International Conference on*, volume 1, pages 682–689. IEEE, 2014.

[45] Conrad Sanderson, Danny Gibbins, and Stephen Searle. On statistical approaches to target silhouette classification in difficult conditions. *Digital Signal Processing*, 18(3):375–390, 2008.

[46] Patricia A. Feineigle, Daniel D. Morris, and Franklin D. Snyder. Ship recognition using optical imagery for harbor surveillance. *Proceedings of Association for Unmanned Vehicle Systems Inter-national (AUVSI), Washington, DC*, 2007.

[47] Abdullah Ibrahim A. Alfadda. *Temporal Frame Difference Using Averaging Filter for Maritime Surveillance*. PhD thesis, Virginia Tech, 2015.

[48] Fouad Bousetouane and Brendan Morris. Off-the-Shelf CNN features for fine-grained classification of vessels in a maritime environment. In *International Symposium on Visual Computing*, pages 379–388. Springer, 2015.

[49] Sergiy Fefilatyev. *Detection of marine vehicles in images and video of open sea*. PhD thesis, University of South Florida, 2008.

[50] Wolfgang Krüger and Zigmund Orlov. Robust layer-based boat detection and multi-target-tracking in maritime environments. In *2010 International WaterSide Security Conference*, pages 1–7. IEEE, 2010.

[51] B. P. Welford. Note on a method for calculating corrected sums of squares and products. *Technometrics*, 4(3):419–420, 1962.

[52] Leila Fallah Araghi, Hamid Khaloozade, and Mohammad Reza Arvan. Ship identification using probabilistic neural networks (PNN). In *Proceedings of the international multiconference of engineers and computer scientists*, volume 2, pages 18–20, 2009.

[53] Robert McKee. *Substance, structure, style, and the principles of screenwriting*. New York: HarperCollins, 1997.

[54] Mike Alder. Newton's flaming laser sword. *Philosophy Now*, 46:29–32, 2004.

[55] Carlos Miguel Madrid Casado. A brief history of the mathematical equivalence between the two quantum mechanics. *Latin-American Journal of Physics Education*, 2(2):9, 2008.

[56] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.