AFRL-AFOSR-UK-TR-2014-0006

Modelling and Characterisation of Detection Models in WAMI for Handling Negative Information

Simon Julier

University College London (UCL)
Gower Street
London WC1E 6BT
UNITED KINGDOM

EOARD Grant 12-2142

Report Date: February 2014

Final Report from 31 August 2012 to 30 August 2013

Air Force Research Laboratory
Air Force Office of Scientific Research
European Office of Aerospace Research and Development
Unit 4515, APO AE 09421-4515

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
21 February 2014	Final Report	31 August 2012 – 30 August 2013

4. TITLE AND SUBTITLE

Modelling and Characterisation of Detection Models in WAMI for Handling Negative Information

5a. CONTRACT NUMBER

FA8655-12-1-2142

5b. GRANT NUMBER

Grant 12-2142

5c. PROGRAM ELEMENT NUMBER

61102F

6. AUTHOR(S)

Professor Simon Julier

5d. PROJECT NUMBER

5d. TASK NUMBER

5e. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UNIVERSITY COLLEGE LONDON (UCL)
GOWER STREET
LONDON WC1E 6BT
UNITED KINGDOM

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

EOARD
Unit 4515
APO AE 09421-4515

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/AFOSR/IOE (EOARD)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

AFRL-AFOSR-UK-TR-2014-0006

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The objective of the project reported here was to develop, implement and evaluate a model of the probability of detection of moving objects in Wide Area Motion Imagery (WAMI) that incorporates the effects of the target, the platform, and the environment. Developing situation awareness is vital for almost any kind of military operation. Through understanding the state and nature of the environment, military personnel can plan and respond accordingly. Situation awareness is often treated as the problem of knowing where all the potential targets are. Through knowing the locations of these targets, threats can be identified and countered. Another important source of awareness is to understand where targets cannot be. Regions that are free of targets can be used to constrain where targets might be. To meet these needs, Wide Area Surveillance (WAS) systems have been developed that are able to sense large swaths of an environment simultaneously and at high resolution. However, the next key challenge is to automatically analyze this image data to, for example, track the locations of targets and identify potential anomalous behavior. This report begins to explore how the output from a WAS system can be used by a state-of-the-art multi-target tracker. In particular, we considered how the output of the image processing and matching algorithms used in the Likelihood of Features Tracker (LoFT) could be combined with a Probabilistic Hypothesis Density (PHD) Filter. Using machine learning techniques, we have developed a formalism and algorithms to automatically predict how the visual appearance of a vehicle can change over time. Using this prediction model, we are then able to automatically threshold and detect potential candidate vehicle locations, and assess both probability of detection and the probability of clutter.

15. SUBJECT TERMS

EOARD, video analysis, cyber security

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18, NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			James H Lawton, PhD
UNCLAS	UNCLAS	UNCLAS	SAR	41	19b. TELEPHONE NUMBER *(Include area code)* +44 (0)1895 616187

Report on "Modelling and Characterisation of Detection Models in WAMI for Handling Negative Information"

Simon J. Julier[1], Amadou Gning[2], Luke Teacy[3], Kannappan Palaniappan[4], Rengarajan Pelapur[5]

February 21, 2014

[1]Department of Computer Science, University College London, UK, s.julier@ucl.ac.uk
[2]Department of Computer Science, University College London, UK, e.gning@ucl.ac.uk
[3]Department of Electronics and Computer Science, University of Southampton, Southampton, UK, wtlt@ecs.soton.ac.uk
[4]Computer Science Department, University of Columbia-Missouri, Columbia, MO, USA, pal@missouri.edu
[5]Computer Science Department, University of Columbia-Missouri, Columbia, MO, USA, rengarajanpelapur@mail.missouri.edu

Executive Summary

Developing situation awareness is vital for almost any kind of military operation. Through understanding the state and nature of the environment, military personnel can plan and respond accordingly. Situation awareness is often treated as the problem of knowing where all the potential targets are. Through knowing the locations of these targets, threats can be identified and countered. To meet these needs, Wide Area Surveillance (WAS) systems have been developed which are able to sense large swaths of an environment simultaenously and at high resolution. However, the next key challenge is to automatically analyse this image data to, for example, track the locations of targets and identify potential anomalous behaviour.

This report begins to explore how the output from a WAS can be used by a state-of-the-art multi-target tracking system. In particular, we considered how the output of the image processing and matching algorithms used in the Likelihood of Features Tracker (LoFT) could be combined with a Probabilistic Hypothesis Density (PHD) Filter. Using machine learning techniques, we developed a formalism and algorithms to automatically predict how the visual appearance of a vehicle can change over time. Using this prediction model, we are then able to automatically threshold and detect potential candidate vehicle locations, and assess both probability of detection and the probability of clutter.

To test the performance of this approach, the machine learned feature prediction model was combined with a PHD filter and applied on several WAS reference datasets. The results were quantified in terms of track duration and integrity, and substantial performance benefits were obtained. We also discuss potential future developments of these algorithms.

Contents

Chapter 1

Introduction

1.1 Motivation

1.1.1 Situation Awareness

Developing situation awareness is vital for almost any kind of military operation [2]. Through understanding the state and nature of the environment, military personnel can plan and respond accordingly. Situation awareness is often treated as the problem of knowing where all the potential targets *are*. Through knowing the locations of these targets, threats can be identified and countered. Another important source of awareness is to understand where targets *cannot be not*. Regions which are free of targets can be used to constrain where targets might be [5, 12]. Furthermore, regions without targets can be of direct tactical importance in their own right. For example, they can be used to plan egress routes.

Given these operational considerations, Wide Area Surveillance (WAS) offers an important solution. Through the use of sensors with a wide field of view, large swathes of the environment can be monitored simultaneously. This makes it possible to identify individual targets and groups of targets which can constitute potential risks.

An important approach for conducting WAS is to use Wide Area Motion Imagery (WAMI) [7].

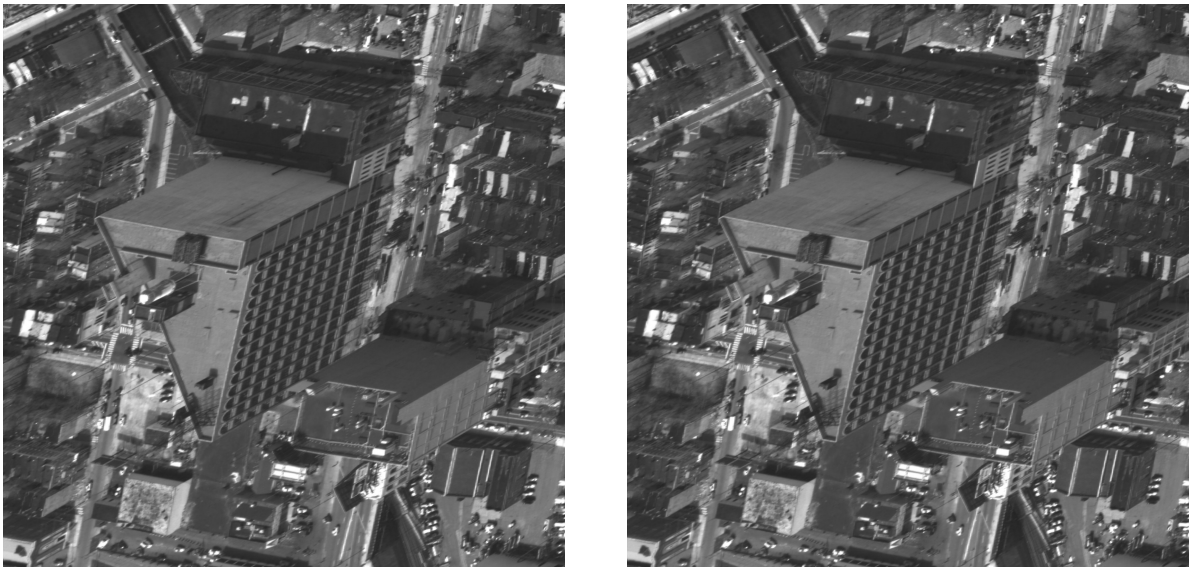
1.1.2 Wide Area Motion Imagery to Conduct Wide Area Surveillance

An urban environment is monitored using an airborne camera sensing array with a high spatial resolution, low frame rate (one to ten frames per second) imaging system. Using such a high resolution image, large numbers of targets can be detected and tracked simultaneously. Figure 1.1 shows an example of the imagery which can be collected by WAMI systems.

The single frame provides a detailed, high resolution view of a large part of the environment. Targets such as individual vehicles are visible for many frames. However, tracking in such images is extremely challenging for many reasons. These include the relatively small size of targets, the large number of targets, and changes in appearance due to changes in environmental conditions and the relative attitude between the target and the camera.

To meet these challenges, the problem has undergone intense research and development and a number of different systems have been developed. For thework carried out here, we are using the Likelihood of Features Tracker (LoFT) [7–9]. LoFT is an appearance-based tracking system. Initialised by an operator, LoFT attempts to track a moving vehicle through a sequence of images. The most significant difficulty in this process is the ability to consistently identify and track the same target through subsequent frames. To improve robustness, a range of image based features are used. These include gradient orientation information using histogram of oriented gradients, gradient magnitude, intensity maps, median binary patterns and shape indices based on eigenvalues of the Hessian matrix.

Figure 1.1: A frame from the Four Hills dataset. This single frame includes images of a large number of cars and buildings. This data set is used extensively in our preliminary investigation.



(a) Frame 46905. (b) Frame 47022.

Figure 1.2: Two frames from a WAMI sequence. Note that the different vantage points of the camera mean that different parts of the street are visible at different times. From [7].

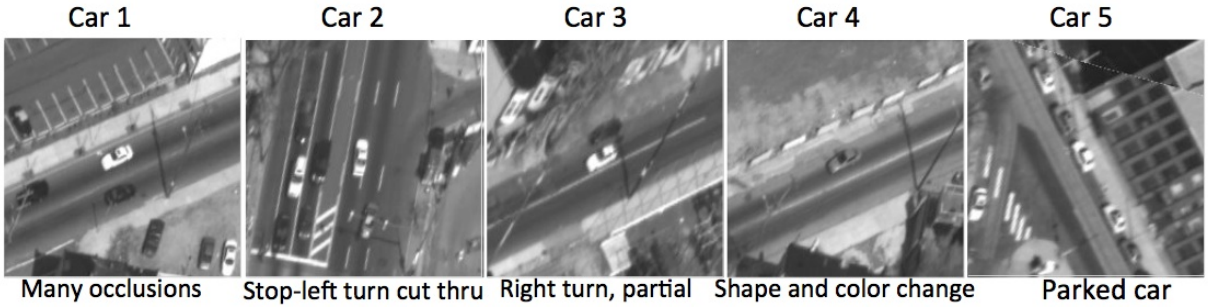


Figure 1.3: Some of the challenges associated with tracking vehicles even when they lie within the field-of-view of the camera. From [7].

Although LoFT provides extremely important capabilities in tracking targets, LoFT there a number of limitations:

1. **Operators initiate the tracks.** This supports a concept of operation in which a target vehicle, believed to be of interest, is to be followed. However, this does not support the notion of a sensing system which will automatically create situation awareness, particuarly in large, complicated environments with many targets.

2. **Track loss can still occur.** Track loss can occur for a variety of reasons. These are principally caused by unmodelled changes in appearance. However, they are also caused by

3. **The system only works with positive returns.** The tracking system works with explicit detections of targets. As such, it cannot exploit information about lack of detections — including the effects of occlusion.

To investigate these issues, we are beginning to explore how LoFT's sophisticated image processing

algorithms can be combined with state-of-the-art multi-target tracking algorithms. In particular, we use machine learning techniques to model the behaviour of the multi-stage detectors used in LoFT. This model is then used in a *Probabilistic Hypothesis Density Filter* (PHD). Unlike most multitarget tracking algorithms which create and propagate a discrete set of tracks, a PHD filter propagates *target intensity*. When integrated over a region of state space, it provides an estimate of the average number of targets which can be found in that regin. As such, it can support complicated multimodal distributions and arbitrary detection models, including regions where no observations can be made at all.

The PHD describes its observation process through the use of a "pseudo-likelihood". In particular, terms for the clutter, the measurment likelihood and the probability of detection must be specified. Because of the complexity of image processing algorithms, simple, closed form solutions for the terms in these equations cannot be derived. Therefore, we decided to use machine learning techniques which could model — and predict — the behaviour of the detectors in LoFT. Because we are using function approximation techniques, and because it is not possible to exhaustively collect data in all possible operating conditions, we use a function approximation approach known as a *Gaussian Process* (GP). GPs are a method of function approximation in which the estimated function value also includes an explicit estimate of the accuracy of the approximation.

1.2 Structure of the Report

The structure of the report is as follows. The next chapter introduces the background on Wide Area Motion Imagery (WAMI) and describes LoFT in greater detail. Multi-target tracking using a PHD filter is described in Chapter 3. Chapter 4 introduces the adaptive framework for observation modelling that we use. It introduces the GP, describes how it was trained, and presents some preliminary results based on simple track likelihood experiment. The full integration of the GP filter and the PHD filter is work in progress. Chapter 5 describes the implementation. The summary is presented in Chapter 6.

Chapter 2

Problem Statement

2.1 Introduction

This chapter introduces the motivation behind the research and describes the Likelihood of Features Tracker (LoFT), the tracking system which we use as a base to develop our work. The structure of this chapter is as follows. Section 2.2 introduces Wide Area Motion Imagery (WAMI) and outlines its importance. Section **??** presents a mathematical model of it. Section 2.3 introduces LoFT and discusses its strengths and weaknesses. We conclude by identifying potential areas of contribution by this work.

2.2 Wide Area Surveillance

2.2.1 Motivation

Situation awareness is critical for many military operations. One way to achieve this is actively by pointing high resolution sensors at targets or areas of interest. However,

Wide Area Surveillance (WAS): the environment is continuously monitored by a sensing system

One source of data is the wide-area large format (WALF) video that is airborne imagery characterized by large spatial coverage, high resolution of about 25 cm GSD (Ground Sampling Distance) and low frame rate of a few frames per second. Wide-area large format imagery is also known by several other terms including wide-area aerial surveillance (WAAS), wide-area persistent surveillance (WAPS), Large Volume Streaming Data (LVSD) and wide-area motion imagery (WAMI) [1, 4, 6, 7].

Tracking in such imagery is challenging as the objects of interest are only 100 square pixels, have seemingly large changes in motion due to the low frame-rate, oblique viewing angles of the camera resulting in occlusions from tall structures apart from noise in the images which could be the result of inaccuracies in flight path or due to atmospheric conditions. Wide-Area video can help determine normal as well as anomalous traffic patterns especially in complex urban environments where persistent tracking of an object is challenging due to the scene content alone. Due to it's wide field of view and high resolution these images contain large amounts of scene content. Such content needs to be analyzed for events of interest from a safety and security standpoint using an automatic/semi-automatic process.

2.2.2 The Four Hills Dataset

One example of a dataset which is available is the Four Hills dataset. Four images from this dataset are show in Figure 2.1.

Figure 2.1: Sample images from the Four Hills dataset.

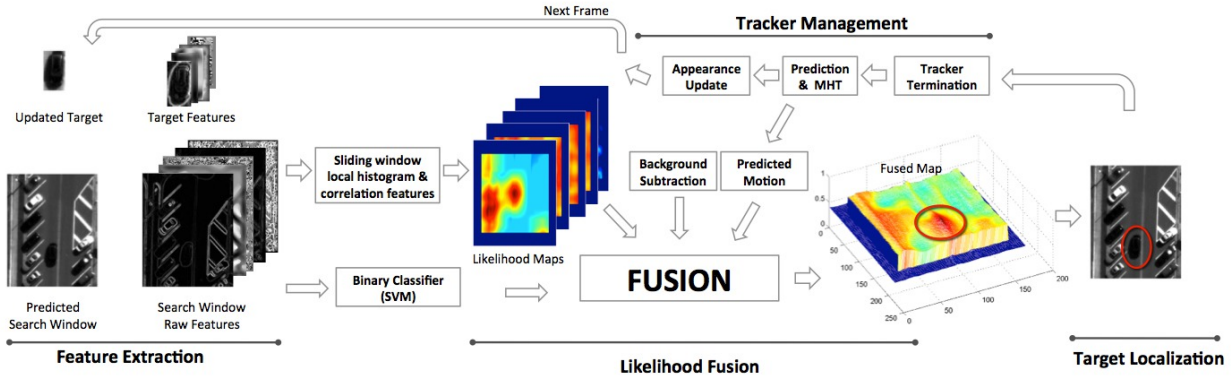


Figure 2.2: The flowchart of LoFT.

2.3 LoFT

2.3.1 Overview

To address these challenges, the LoFT (Likelihood of Features Tracking) system was developed. The flowchart of LoFT is shown in Figure 2.2. By clicking on a frame, an operator manually selects an object which is initiated within the track management system. The pixels immediately around the pixel where the frame is clicked are interrogated, and a visual signature, constructed of several templates, is constructed. A variety of region-based, edge-based, local shape-based, and texture-based classifiers are used to improve the robustness of the tracker. The features we use in this project are listed in Figure 2.4. The LoFT system has been developed over a great deal of time, and includes a great deal of work on feature detectors [8], motion models [13] and descriptor and template adaptation [9].

2.3.2 State Model

The state space of LoFT is defined in 2D pixel coordinates and consists of the position and velocity of the target together with the orientation of the template,

$$\mathbf{x}_k = \begin{bmatrix} r_k \\ c_k \\ \dot{r}_k \\ \dot{c}_k \\ \theta_k \end{bmatrix} \tag{2.1}$$

where r_k, c_k are the rows and columns of the images, $\dot{r}_k$ and $\dot{c}_k$ are the column velocities. θ_k is the orientation of the stored template.

The position is assumed to evolve using a piecewise constant velocity model in pixel space. The orientation is assumed to remain constant. Therefore, the process model is

$$r_{k+1} = r_k + \Delta T_k \dot{r}_k \tag{2.2}$$
$$c_{k+1} = c_k + \Delta T_k \dot{c}_k \tag{2.3}$$
$$\dot{r}_{k+1} = \dot{r}_k \tag{2.4}$$
$$\dot{c}_{k+1} = \dot{c}_k \tag{2.5}$$
$$\theta_{k+1} = \theta_k. \tag{2.6}$$

This model can adequately describe the motion of a 2D template. However, it does not directly account for the motion of the camera or occlusion effects in the environment. To account for this, each image is

Feature	Name	Dimension
Intensity Histogram	`hist_I`	10
Gradient Histogram	`hist_M`	10
ARST Histogram	`hist_A1`	10
SI Histogram	`hist_A2`	10
NC Histogram	`hist_VH`	10
HoG Histogram	`hist_HOG`	10
Intensity Block	`corr_I`	2500
Gradient Block	`corr_M`	2500
Total	—	5060

Table 2.1: The features used and their dimensions.

registered with respect to the first frame. Figure 2.3 shows a sequence of registered images. Although this superficially appears to produce a highly stable image, there are issues with

2.3.3 Observation Models

LoFT performs feature fusion by comparing a target appearance model within a search region using feature likelihood maps which estimates the likelihood of each pixel with the search window belonging to part of the target [9]. Because LoFT is initialised manually, it does not use detectors to identify the potential presence of vehicles. Rather, given a first frame, the system constructs a visual signature which can be used to explain how appearance evolves over time. Specifically, suppose a target is present in the image with $\mathbf{x}_k$. The entire vehicle is assumed to be bounded within the rectangular region $\mathbf{r}_k = \langle r_k, c_k, w_k, h_k \rangle$, where r_k and c_k are from target state, and w_k and h_k are the width and height of the region.

Given $\mathbf{r}_k$, the target's appearance at time k is described by signature, which consists of a set of *features*, $\mathbf{f}_k \in \mathcal{F}$, where each element of $\mathbf{f}_k$ is a measurement of some characteristic of the pixels bounded by $\mathbf{r}_k$. Figure 2.4 illustrates some of the features used. These include mean colour intensities, and Histograms of Oriented Gradiants (HoG). Table 2.1 summarises the dimensions associated with the features. Although some of these are relatively low-dimensional, the correlation blocks are very large and the overall dimension of the feature vector is 5060.

In many situations, a fixed template is not sufficient to ensure robust tracking. However, the decision of when to update the template is known as the stability-plasticity dilemna: if the changes are too frequent, the template can capture subtle tracking errors, occlusions and change in lighting. If it happens too infrequently, tracks will be lost. The way this is achieved in LoFT is that, at each frame, LoFT attempts to estimate the current rotation of the template. If this exceeds θ_k by a fixed threshold, a new template is computed from the image and θ_k is replaced by the new template angle.

In the preliminary work undertaken here, we do not use template adapation. Rather, the features are fixed in the first frame, and changes are learned in subsequent frames. We will investigate the use of this later.

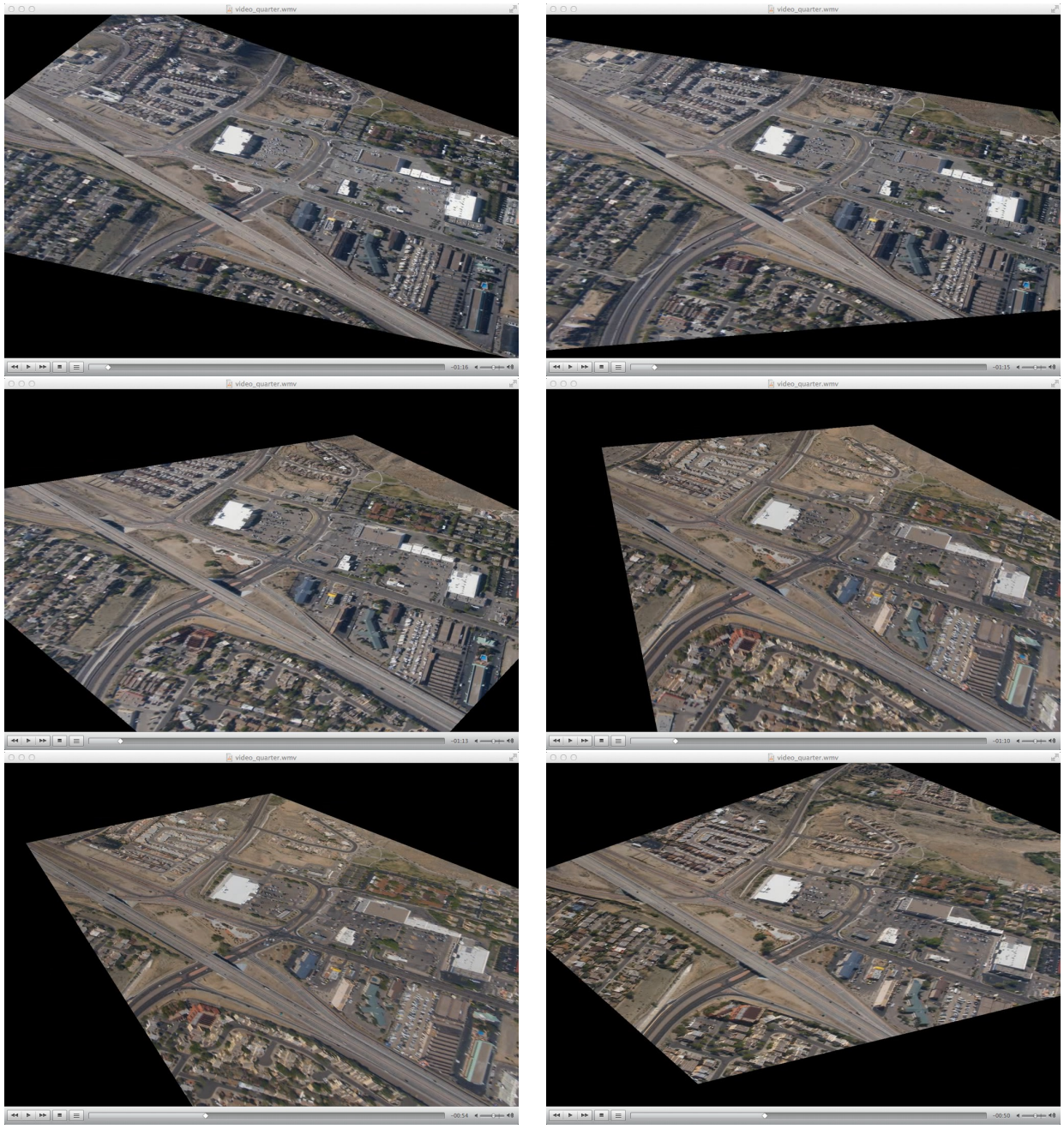


Figure 2.3: A sequence of registered frames in the Four Hills dataset. These are registered with respect to feature descriptors associated with the ground plane.

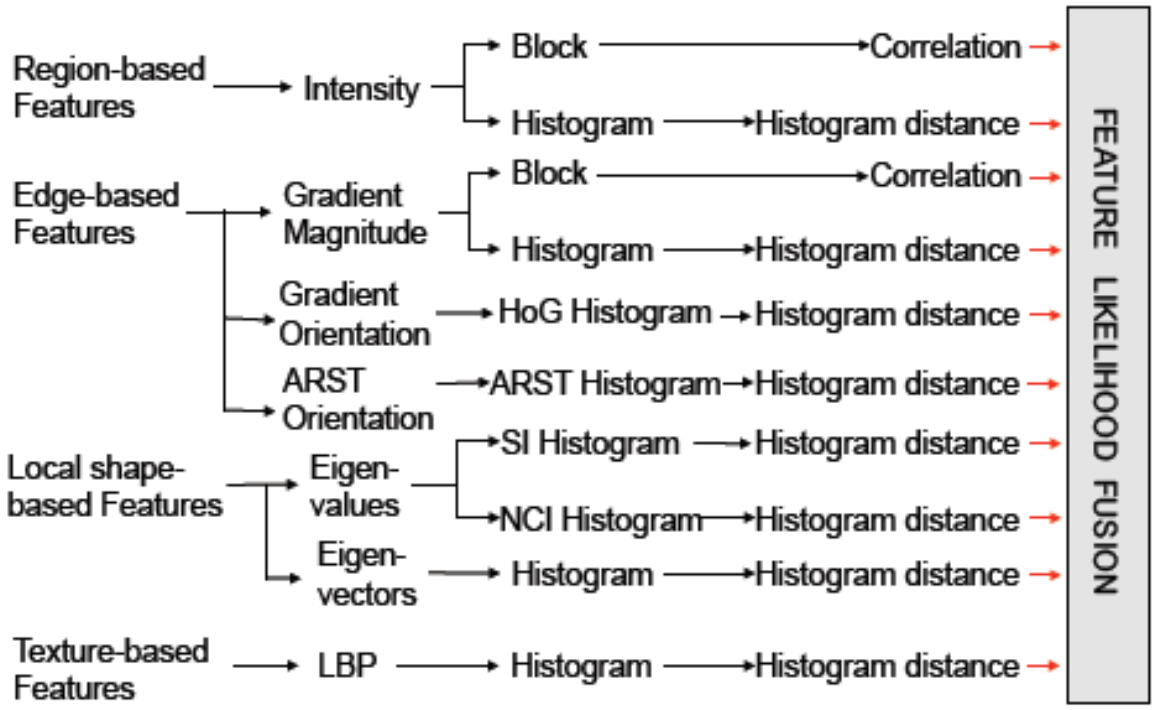


Figure 2.4: LoFT features.

Chapter 3

Multi-target Tracking

3.1 Introduction

The challenges posed by WAMI can be viewed in terms of an example of multi-target tracking. In this chapter, we first lay out the formulation of multi-target tracking using random finite sets. We then describe the PHD filter formulations.

3.2 A Finite Set Statistic Approach to Multi Object Tracking

Mahler argued that the correct way to consider the problem of multi-object tracking is to use random sets. A random set is a generalisation of a vector-valued random variable to the case where the number of random variables is not known. It can be used to represent both the distribution of objects in the environment, and the observations received from a sensor.

Suppose at time k there are $N(k)$ targets, each one taking a value in a state space $\mathcal{X}$. The state of the environment, X_k can be written as the set

$$X_k = \left\{\mathbf{x}_{k,1}, \ldots, \mathbf{x}_{k,N(k)}\right\} \subset \mathcal{X}. \tag{3.1}$$

This is a *random* set — both the cardinality $N(k)$ and the state of each target is unknown and must be estimated.[1]

The evolution of the state is described by the following equation,

$$\Xi_{k+1|k} = \Gamma\left(X_k\right) \cup B\left(X_k\right) \cup B, \tag{3.2}$$

where $\Gamma\left(X_k\right)$ describes the evolution of the persistent targets, $B\left(X_k\right)$ is the set of spawned targets and B are the set of additional targets generated independently of the existing targets.

The idea is that a target survives with a probability $P_S\left(\mathbf{x}_{k,i}\right)$. If it survives, then it evolves using the standard process model.

The environment is observed by camera affixed to an airborne platform. We assume, for simplicity, that the pose of the camera is measured by an extremely accurate external sensing system. As a result, we assume that the pose of the camera is perfectly known and is given by the state vector $\overset{*}{\mathbf{x}}_k$.

As explained in Subsection 2.3.3, each frame in the camera is processed using LoFT's detection system. This yields a set of detections together with specified pixel coordinates. Suppose that $M(k)$ detections are acquired. These are collected into the observation set.

$$Z_k = \left\{\mathbf{z}_{k,1}, \ldots, \mathbf{z}_{k,M(k)}\right\} \subset \mathcal{Z}. \tag{3.3}$$

[1] An important property of the set is that the *order* of the elements does not matter. Therefore, X_k is equivalent to the set $\left\{\mathbf{x}_{k,N(k)}, \ldots, \mathbf{x}_{k,1}\right\}$. In consequence, there is no strict ordering between the location of an element in the set and the target ID.

This, too, is treated as a random set.

The measurement model which underlies this set of detections is given as follows. The measurement set is given by

$$\Sigma_k = \Gamma(X_k, \overset{*}{\mathbf{x}}_k, \mathbf{e}) \cup C(\overset{*}{\mathbf{x}}_k, \mathbf{e}). \tag{3.4}$$

$\Gamma(X)$ is the target detection set and $C(X)$ is the set of false detections. Suppose the environment consists of n targets, and the state of the ith target is $\mathbf{x}_i$. The target detection set is of the form

$$\Gamma(X) = \Gamma(\mathbf{x}_{k,1}) \cup \ldots \cup \Gamma(\mathbf{x}_{k,N(k)}) \tag{3.5}$$

where

$$\Gamma(X_k, \overset{*}{\mathbf{x}}_k, \mathbf{e}) = \emptyset^{P_D(X)} \cap \{\mathbf{Z}_i\} \tag{3.6}$$

means that the target measurement is detected with a probability of $P_D(X) = p_D(X_k, \overset{*}{\mathbf{x}}_k, \mathbf{e})$. The clutter process $C(\overset{*}{\mathbf{x}}_k, \mathbf{e})$ is given by

$$C(\overset{*}{\mathbf{x}}_k, \mathbf{e}) = C(\overset{*}{\mathbf{x}}_k, \mathbf{e}). \tag{3.7}$$

These equations require some further justification. The observation of an actual target depends upon the relative pose between the platform and the target. The probability of detection depends, in general, upon the relative configuration of the target as well. We expect that, in general, detection algorithms are likely to be more successful in some configurations than others. The environment has an impact on this as well — in some environments a vehicle is likely to be easier to be seen than in others. Similarly, the clutter is generated by elements of the environment — such as parked vehicles or rubbish bins — which can appear like cars.[2]

However, although the RFS provides a very general framework for tackling multi-object tracking problems, many of the algorithms have factorial complexity and thus have little advantage over previous approaches. Therefore, a new representation is required.

3.3 Approximate Multi-Object Tracking Through the Use of PHD Filters

The fundamental reason why multi-object tracking becomes challenging is that, as the number of targets increase, the complexity of the representation of the environment and the computational complexity rises. Therefore, one way to address the problem is to derive a way of representing the number of targets in a way that the complexity does not increase with the number of targets. The way to achieve this is through propagating the *target density*.

The intution behind this approach is illustrated in Figure 3.1. The figure illustrates a typical multitarget tracking example and shows the intensity.

More formally, the intensity is the first moment of the random finite set X_k statistic called the *Probability Hypothesis Density* (PHD). Let define $D\left(\mathbf{x}|Z^k\right)$ as the PHD associated with the multi-object posterior $p(X_k|Z^k)$ at a time step k. The intensity has the property that

$$\mathrm{E}\left[|\Xi \cup \mathcal{R}|\right] = \int_{\mathcal{R}} D\left(\mathbf{x}|Z^k\right)\mathrm{d}\mathbf{x}. \tag{3.8}$$

It is important to note that the PHD is *not* the same as a probability distribution. The easiest way to see this is that integrating the PHD over the entire state space yields the *expected number of targets*.

The important practical advantage of the use of the PHD is that, given a number of assumptions, compact closed form solutions can be derived which have a computational cost which is *linear* in the number of observations.

[2]We do not assume that the clutter is target state dependent because it is not clear how target-dependent clutter would be generated in this scenario.

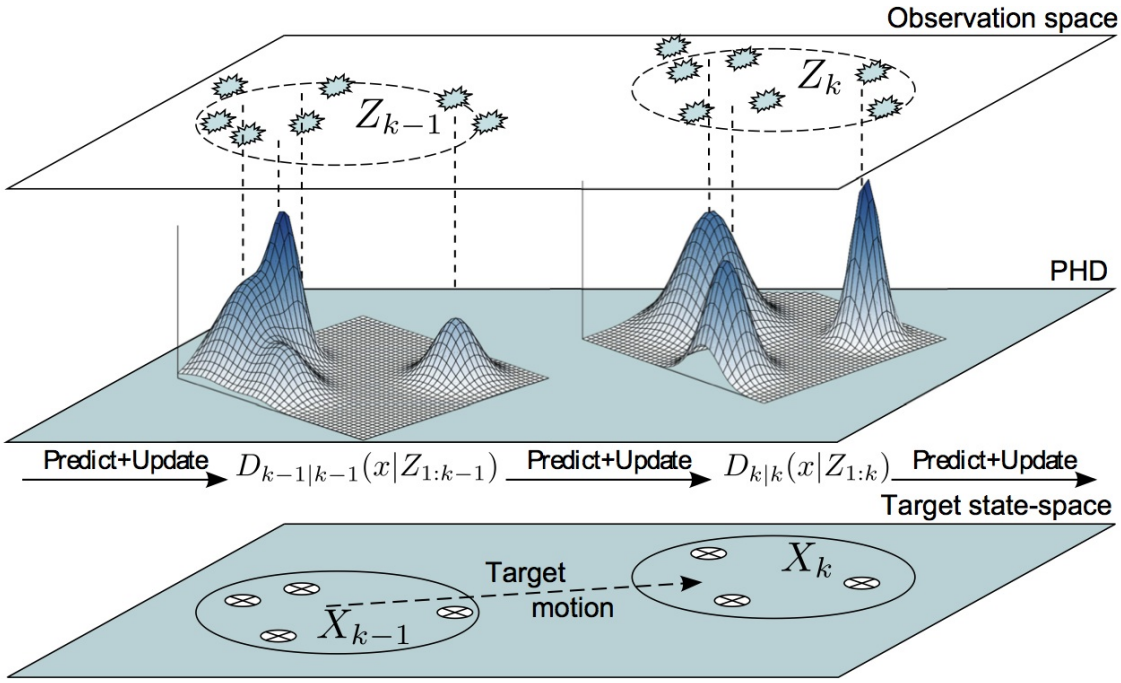


Figure 3.1: Illustration of the PHD filter. The bottom figure shows the evolution of the state in terms of a random set and illustrates the time evolution, including the disappearance of a target. The top figure shows the pattern of observations, including the presence of clutter. The middle figure shows the intensity representation. Peaks in intensity show where the average number of targets is greatest.

3.3.1 PHD Filtering Equations

The PHD filter utilises Random finite set statistics to sequentially propagate the intensity functions through the Bayesian steps. The main advantage of the PHD filter is a formulation without a data association. Instead the intensity function is updated with the random set of measurements Z_k as it is shown below.

The prediction can be realized through the following equation:

$$D(\mathbf{x}_k|Z^{k-1}) = b(\mathbf{x}_k) + \int p_s(\mathbf{x}_{k-1})p(\mathbf{x}_k|\mathbf{x}_{k-1})D(\mathbf{x}_{k-1}|Z^{k-1})\mathrm{d}\mathbf{x}_{k-1}, \tag{3.9}$$

where $b(\mathbf{x}_k)$ denotes the intensity function of spontaneous birth of new objects, $p_s(\mathbf{x}_{k-1})$ is the probability that the object still exists at the time step k given its previous state $\mathbf{x}_{k-1}$, and $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the transition probability density of the individual objects.

For the update model, it is assumed that the false alarms obey the following conditions. First, the average number of clutter detections is Poisson distributed with a mean $\lambda = \lambda_{k+1}(\overset{*}{\mathbf{x}}_k, \mathbf{e})$ false alarms. Second, the spatial distribution of these clutter terms is given by $c(\mathbf{z}) = c_{k+1}(\overset{*}{\mathbf{x}}_k, \mathbf{e})$. It is further assumed that the predicted multitarget distribution is approximately Poisson.

Given these assumptions, the update equation can be written as

$$D\left(\mathbf{x}_k|Z^k\right) \cong L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k)D\left(\mathbf{x}_k|Z^{k-1}\right), \tag{3.10}$$

where $L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k)$ is the PHD pseudolikelihood. Its value is given by

$$\begin{aligned} L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k) &= 1 - p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e}) + \\ & p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e}) \sum_{\mathbf{z}\in Z_k} \frac{L_{\mathbf{z},\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k)}{\lambda c(\mathbf{z}|\overset{*}{\mathbf{x}}_k, \mathbf{e}) + \int p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e}) L_{\mathbf{z},\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k) D\left(\mathbf{x}_k|Z^{k-1}\right)\mathrm{d}\mathbf{x}_k}, \end{aligned} \tag{3.11}$$

where $p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e})$ is the probability that the sensor, with state $\overset{*}{\mathbf{x}}_k$ flying over environment $\mathbf{e}$ is able to detect a target whose state is $\mathbf{x}_k$, $L_{\mathbf{z},\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k)$ is the likelihood of $\mathbf{x}_k$ given observation $\mathbf{z}$, λ is the average number of clutter points per scan and $c(\mathbf{z}|\overset{*}{\mathbf{x}}_k, \mathbf{e})$ is the probability of the clutter return $\mathbf{z}$.

3.3.2 Particle Filter-Based Implementation

In general, the probability distribution can be hard to write down. Therefore, in this work we use a Sequential Monte Carlo (SMC) based implementation of the PHD filter.

The SMC implementation approximates the PHD by a weighted set of N_k particles,

$$D\left(\mathbf{x}_k|Z^k\right) \approx \sum_{i=1}^{N_k} w_{k|k}^{(i)} \delta\left(\mathbf{x}_{k|k}^{(i)} - \mathbf{x}_k\right), \tag{3.12}$$

where $\delta(\cdot)$ is the vector form of a delta function and

$$\sum_{i=1}^{N_k} w_{k|k}^{(i)} = \eta_{k|k}, \tag{3.13}$$

which is the expected number of targets.

The SMC-PHD filter consists of the following steps:

1. **Predict target intensity.** This consists of two steps: predict existing particles forwards, and modelling the spontaneous birth of targets.

(a) *Predicting existing particles forwards.* The process model is applied to each particle $\mathbf{x}_{k|k}^{(i)}$ to generate a predicted particle $\mathbf{x}_{k+1|k}^{(i)}$. The weights on the particles are unchanged, and so $w_{k+1|k}^{(i)} = w_{k|k}^{(i)}$.

(b) *Target birth.* To model $b(\mathbf{x}_k)$, we use the approach proposed in [10]. The idea is that, around each new measurement, a new set of particles are created. The state of each particle is initialised by inverting the observable parts of the state and sampling uniformly over the parts which aren't observable. A total of $N_{k,new}$ particles are created this way. Each particle receives a uniform weight equal to the number of new particles divided by the expected number of targets which appear at each time step.

The result of these two steps is that the predicted PHD is of the form

$$D\left(\mathbf{x}_{k+1}|Z^k\right) \approx \sum_{i=1}^{N_k+N_{k,new}} w_{k+1|k}^{(i)} \delta\left(\mathbf{x}_{k+1|k}^{(i)} - \mathbf{x}_{k+1}\right). \tag{3.14}$$

2. **Compute correction term.** For each measurement $\mathbf{z}_{k+1,i}$ in Z_{k+1}, compute the term

$$\lambda_{k+1|k}\left(\mathbf{z}_{k+1,i}\right) = \lambda c\left(\mathbf{z}_{k+1,i}|\overset{*}{\mathbf{x}}_k, \mathbf{e}\right) + \sum_{i=1}^{N_k+N_{k,new}} w_{k+1|k}^{(i)} p_D\left(\mathbf{x}_{k+1|k}^{(i)}|\overset{*}{\mathbf{x}}_k, \mathbf{e}\right) L_{\mathbf{z}_{k+1,i},\overset{*}{\mathbf{x}}_k,\mathbf{e}}\left(\mathbf{x}_{k+1|k}^{(i)}\right). \tag{3.15}$$

The important thing to note is that the correction term is a function of the clutter for each observation, together with the fact that the likelihood and probability of detection are computed for each particle separately.

3. **Update.** The update corresponds to rescaling the particles by a particle form of the PHD pseudolikelihood. Specifically,

$$w_{k+1|k+1}^{(i)} = L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_{k+1|k}^{(i)}) w_{k+1|k}^{(i)} \tag{3.16}$$

where $L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_{k+1|k}^{(i)}) w_{k+1|k}^{(i)}$ is the particle form of the PHD pseudo-likelihood. Its value is given by

$$L_{Z_k,\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_{k+1|k}^{(i)}) = 1 - p_D\left(\mathbf{x}_{k+1|k}^{(i)}|\overset{*}{\mathbf{x}}_k, \mathbf{e}\right) + \sum_{i=1}^{M(k)} \frac{L_{\mathbf{z}_{k+1,i},\overset{*}{\mathbf{x}}_k,\mathbf{e}}\left(\mathbf{x}_{k+1|k}^{(i)}\right) p_D\left(\mathbf{x}_{k+1|k}^{(i)}|\overset{*}{\mathbf{x}}_k, \mathbf{e}\right)}{\lambda_{k+1|k}\left(\mathbf{z}_{k+1,i}\right)}. \tag{3.17}$$

4. **Resample.** As with all SMC implementations, resampling is required to mitigate the effects of particle depletion. The average number of targets is computed from

$$\eta_{k+1|k} = \sum_{i=1}^{N_k} w_{k|k}^{(i)}. \tag{3.18}$$

Any standard particle scheme can be used to resample the number of particles. Once the particles have been resampled, the weights are multipled by $\eta_{k+1|k}$ to ensure that the average number of targets remain the same.

3.3.3 Numerical Example

We consider the problem of tracking one vehicle of interest initiated by an operator at time $k = 0$. When using LOFT alone, as discussed previously, track loss can occurs and in this example we aim to illustrate how a standard PHD combined with LOFT detections offer a simple solution. Instead of returning the best match, LOFT is modified to return a set of detections (see Figure 3.3 consistent with the 90% score of the best match). By doing so, the problem now contains false alarm and misdetections which are well within the PHD framework.

Given the position of the vehicle in pixel coordinates, a bounding box around the vehicle is extracted. Next, a set of features are extracted from this template to initialise LOFT for the first image frame. The PHD-PF filter described in Section **??** is initialised with a set of particles around the starting pixel location, and the sum of the weights of the particles is set to 1 since only a single target is considered as detected. With respect to the PHD filter described in Section 3.3.1 and its PF implementation in Section **??**, the following simplifications and assumptions are made:

- the state is defined in the image space and is constituted of 2 coordinates, *i.e.*, $\mathbf{x}_k$ is a 2D pixel coordinates.
- $N_k = 2000$ particles are used at each step.
- The evolution model is kept simple and independent of the camera's state $\overset{*}{\mathbf{x}}_k$. A random walk evolution model is used with a Gaussian distribution chosen to be of 0 mean with a standard deviation equal to 60 pixels.
- Since we only want to track one vehicle initiated by the operator, the birth process step is eliminated, *i.e.* $b(\mathbf{x}_k) = 0$ and $N_{k,new} = 0$ in (**??**).
- The likelihood calculation is independent of the camera position and the environment, *e.g.*, the probability of detection is not state dependent and is chosen $p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e}) = 0.99$ and the clutter distribution is a uniform Poisson process with parameters $\lambda = \lambda_{k+1}(\overset{*}{\mathbf{x}}_k, \mathbf{e})$ equal to the size of the image frames and $c(\mathbf{z}) = c_{k+1}(\overset{*}{\mathbf{x}}_k, \mathbf{e}) = 5$.

Figure 3.3 shows a set of 14 frames where a vehicle of interest is tracked using LOFT alone and by extracting the set of detections that are consistent with 90% of the best score. While we can manually verify that the vehicle of interest is always tracked the set of candidates grows and it is difficult for an operator to interpret the output.

While the PHD implementation described here is very simple especially with a minimal evolution model the results shown in Figure 3.1 are encouraging. In comparison to the loft detections it can be seen that the PHD filters most of the candidates and, at worst, displays a multimodal distribution. on that regard, it offers a better readability for the operator.








Figure 3.2: Example of LOFT detection on 14 successive frames using a template initiated by an operator

image 2 PHD targets

image 3 PHD targets

image 4 PHD targets

image 5 PHD targets

image 6 PHD targets

image 7 PHD targets

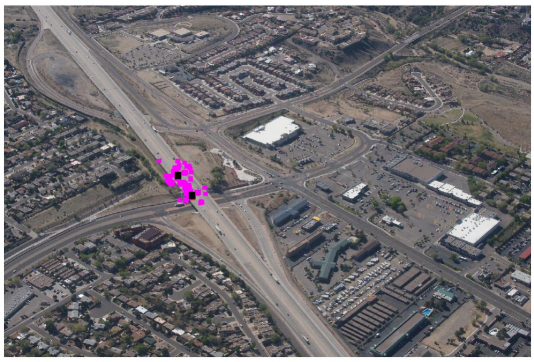


image 8 PHD targets

image 9 PHD targets

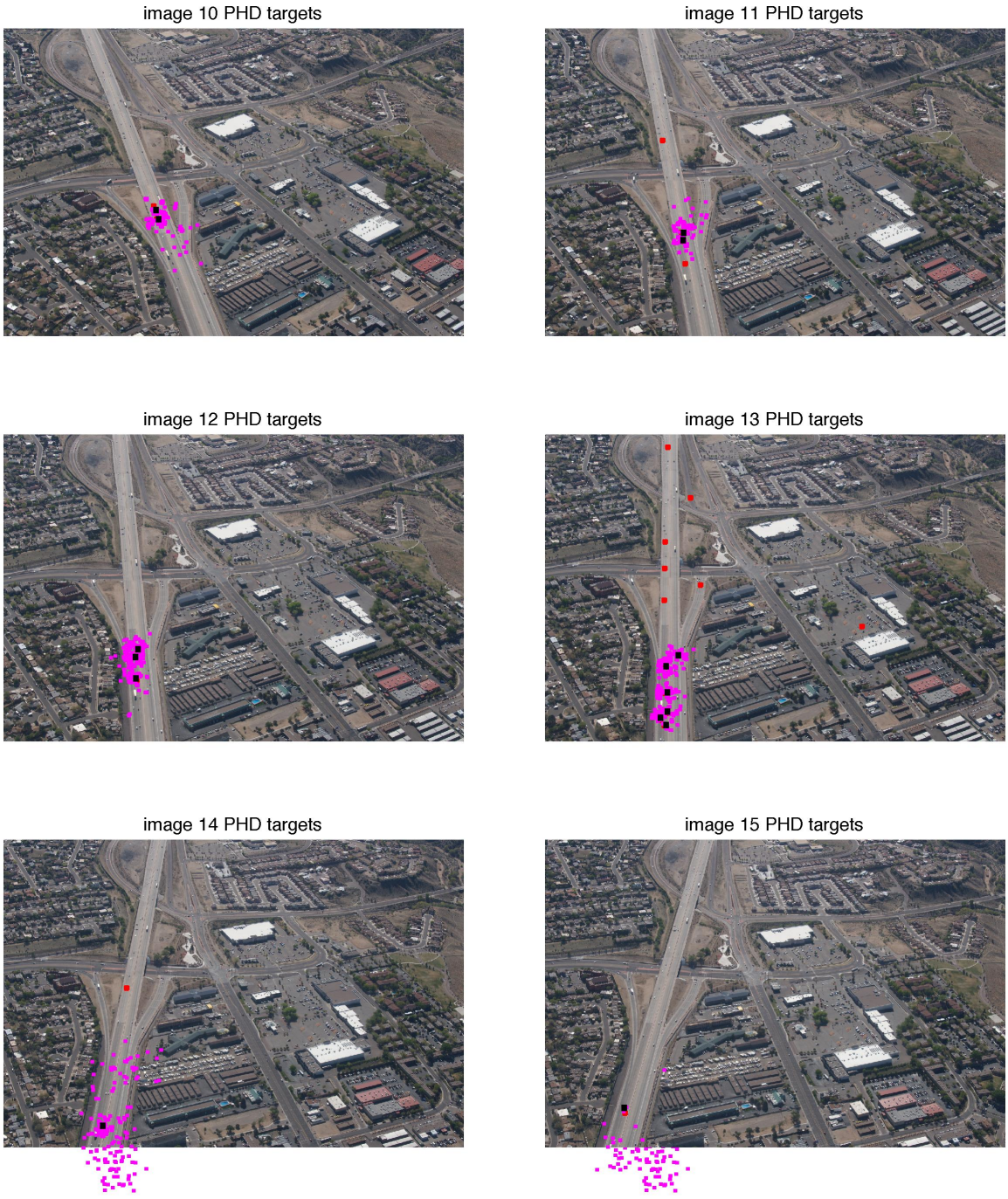


Figure 3.3: Example of LOFT detection on 14 successive frames using a template initiated by an operator

Chapter 4

Probabilistic Model of the Observations

4.1 Introduction

In this chapter, we describe the approach we have taken to develop the model of the observation process. The observation process is modelled in terms of the PHD pseudolikelihood presented in (3.11) and consists of three terms: the probabilty of detection $p_D(\mathbf{x}_k|\overset{*}{\mathbf{x}}_k, \mathbf{e})$, the measurement likelihood $L_{\mathbf{z},\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x}_k)$ and the clutter process $\lambda c(\mathbf{z}|\overset{*}{\mathbf{x}}_k, \mathbf{e})$. For this preliminary report, we focus on the development of an algorithm to *predict* the future appearance of the vehicle. This prediction underpins both the probability of detection and clutter processes. One element of our ongoing work is to refine this process into the detection and clutter terms.

The structure of this chapter is as follows. Section 4.2 describes the challenges which exist in tracking the visual appearance of the vehicle over time. Section **??** describes Gaussian processes, which are the key theoretical tool we use to achieve prediction process. Section 4.4 describes the feature prediction model and how it was tuned. Section **??** evaluates the performance of the prediction model in terms of its ability to compute total track likelihood for ground truthed tracks in a test set.

4.2 Modelling Changes in Visual Appearance

The idea is as follows. Suppose a template was taken at a time k_t where $k_t < k$. At this point, the target was in the state $\mathbf{x}_{k_t}$ and the platform was in the state $\overset{*}{\mathbf{x}}_{k_t}$. The template is described by the set of features $\mathbf{f}_{k_t}$. The goal is to now try to identify the target at time step k. From the estimated target state, a Region of Interest (ROI) can be computed. This is decomposed into a set of n rectangular blocks. Within the ith block, the feature $\mathbf{f}_k^{[i]}$ is computed.

We *predict* how the feature will appear at time k, $\tilde{\mathbf{f}}_k$ and compute the difference $\delta\mathbf{f}_k = \mathbf{f}_k^{[i]} \boxminus \tilde{\mathbf{f}}_k$. If $\delta\mathbf{f}_k$ is sufficiently small, a detection is generated and an observation is inserted into the observation set (3.3). Once all the blocks have been processed, the set of observations can then be passed to the PHD filter to be updated.

The prediction equation is of the form

$$\tilde{\mathbf{f}}_k = f\left[\mathbf{f}_{k_t}, \mathbf{x}_{k_t}, \overset{*}{\mathbf{x}}_{k_t}, \mathbf{x}_k, \overset{*}{\mathbf{x}}_k, \mathbf{e}\right]. \tag{4.1}$$

However, this function arises implicitly through the use of the many difficult visual descriptors used in LoFT. As such, it is not possible to write an explicit expression down. Rather, we use machine learning techniques to learn an approximation of it. In particular, we use Gaussian Processes.

Figure 4.1: Slight variation in target appearance and background over time.

4.3 Approximating Functions Using Gaussian Processes

Gaussian Processes (GPs) are widely used in probabilistic function approximation. Consider the function

$$y = \mathbf{f}[\mathbf{x}] \tag{4.2}$$

where $\mathbf{x}$ is the input and y is the (scalar) output.[1] The form of $\mathbf{f}[\cdot]$ is not known and is to be approximated empirically. A training set $\mathcal{D}$ of n measurements has been collected, where $\mathcal{D} = \{(\mathbf{x}_i, y_i)|i = 1, 2, \ldots, n\}$. The column of input vectors is aggregated into the *design matrix* X, and the training output y. Suppose the output of the function is to be estimated at some test value $\mathbf{x}_*$. The approximation y_*, is a random variable whose the probability density function given by

$$y_* \sim p(f_*|\mathbf{x}_*, X, \mathbf{y}). \tag{4.3}$$

The reason why y_* is a random variable is because reflects the uncertainty in the implicit estimation of the function. The training and test data enter as *conditioning* random variables.

A GP explicitly assumes that y_* is Gaussian-distributed. Therefore, to fully specify (4.3), expresses for the mean and covariance must be provided. The GP provides a formulation for this through the specification of a *kernel function*, which specifies the second order moment of the approximation computed at two different test values.

Many different choices for kernel functions exist. For our application, we use the widely-adopted squared exponential kernel.

Figure 4.2 illustrates the action of the GP for one feature for basic time. This diagram illustrates an important property of the GP: when the test value $\mathbf{x}_*$ is close to an element in the training set, the prediction error is small. This is reflected by the smaller covariance.. As $\mathbf{x}_*$ moves further from the nearest point in the training set, the covariance gradually increases. This reflects the fact that the value of the function to be approximated is far from any member of the training set, and thus there is considerable uncertainty.

4.4 Design of the Feature Prediction Model

We make three assumptions in the development of the model:

[1] Although GPs can be extended to vector-valued predictions, we use the scalar formulation described here. This is both computationally simpler, and is a better fit for our PCA-compressed feature space, where each feature is assumed to be independent of all other features.

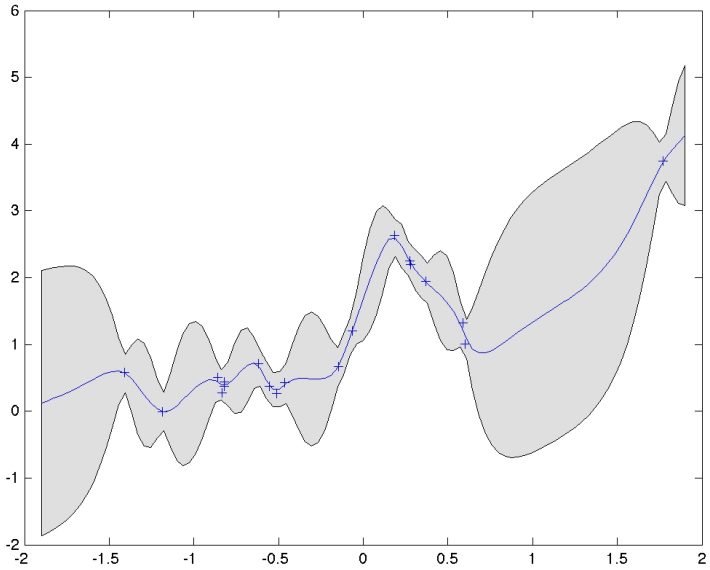


Figure 4.2: A standard example of a GP. The input data (x) and function values (y). The function approximation rises.

1. The mean function can be adequately approximated by a constant.
2. The feature vector is composed of the individual features listed in Table 2.1. We assume that these can be treated as a set of independent scalar features, each of which can be modelled as an independent Gaussian Process.

4.4.1 Dataset

The observation model was tuned using the Four Hills dataset. Some frames from this are shown in Figure 2.1. This was used to both train and test the models. To train the models, a subset of vehicles were manully tracked from frame-to-frame. The test set consists of a different set of models.

Figure 4.3 shows one example of a training set which is used. This consists of approximately 20 frames of a vehicle driving. Variations include changes in orientation and scale. In addition, there is the presence of clutter from other vehicles.

4.4.2 Dimensionality Reduction Using PCA

The set of features that we use from LoFT are listed in Table 2.1. Rather than use these features directly, we apply Principle Component Analysis (PCA). There are two reasons for this. The first is to transform each feature space into one where individual features are uncorrelated. The second is to reduce the number of dimensions by keeping only the dimensions with the most variance, which are likely to be the most informative.

Two criteria were explored in the choice of PCA. The first criteria was to remove dimensions from each feature state until the variance of the reduced set was at least 90% of the original set. The results of this are shown in Table 4.1. This greatly reduces the number of dimensions (from 5060 to 159). However, the block correlation features are still very high dimensional. Therefore, a second criteria was used. In this case, the number of dimensions for each feature was not allowed to exceed 10. The results in Table 4.1 show that this

Figure 4.3: Region of Interest corresponding to target 20 in the Four Hills dataset.

		Unlimited dimensions; 90% variance		Max 10 dimensions; unlimited variance	
Feature	Dimension	Dimension	% Variance	Dimension	% Variance
hist_I	10	2	94%	2	94%
hist_M	10	1	99%	1	99%
hist_A1	10	1	100%	1	100%
hist_A2	10	1	99%	1	99%
hist_VH	10	4	93%	4	93%
hist_HOG	10	4	90%	4	90%
corr_I	2500	30	90%	10	77%
corr_M	2500	116	90%	10	45%
Total	5060	159	—	33	–

Table 4.1: The original number of dimensions in each feature, together with the results of compressing the features using PCA. The first set of columns show the minimum size of feature required when the variance is 90% of the original value. The second set of columns show the effect on the variance when the maximum number of dimensions for each feature is clamped at 10.

is only required for the correlation features and that the approximation in corr_M is much greater than that in corr_I.

4.4.3 Choice of the Dependent Variables

We need to specify the dependent variables which are used in (**??**). Although all the variables listed could be used, this it not preferable for two reasons. The first is that higher dimensional solutions can be expensive and wasteful. The second is that not all the information is available. For example, we are currently using models in 2D. As a result, although bundle adjustment can be used to reconstruct the sequence of values for $\overset{*}{\mathbf{x}}$ (see Section 2.2.2), this information cannot be directly exploited in the 2D formulation.

We considered the following choices for the dependent variables.

Time

This is perhaps the simplest approach. The idea is to model the covariance as a decreasing function of time. The rationale is that, as time progresses, the change in visual appearance will increase through time. Using a squared exponential, the covariance function is of the form

$$k(\mathbf{s}_k, \mathbf{s}'_k) = \sigma^2 \exp\left[-\frac{(t-t')^2}{2l}\right]. \tag{4.4}$$

The rationale behind this choice is that, as time progresses, the changes to the appearance can become greater. Furthermore, all the datasets exhibit a constant angular rotation as a result of how the aircraft is moving. Therefore, time is potentially a proxy for angular change.

Template Orientation

Here, the idea is to look at the rotation of the template directly in image coordinates. This factors in changes due to the rotation of the platform and the turning of the vehicle.

Pixel Coordinates

As above, but replace $\mathbf{s}_k$ with $\langle t, \mathbf{r}_k \rangle$, so we are measuring similarity between image positions and time.

Platform Orientation

This approach extends the covariance to be a decreasing function of the change in viewing angle on the vehicle. We compute this from extracting the camera pose information from bundler, and computing the relative transformations between pairs of terms. As a result, the kernel incorporates the deterministic change in platform orientation caused by the aircraft movement.

$$k(\mathbf{s}_k, \mathbf{s}'_k) = \sigma^2 \exp\left[-\frac{1}{2}\sum_{k=1}^{d}\frac{(s_{k,t} - s'_{k,t})^2}{l_k^2}\right] \tag{4.5}$$

For the straight line roads, the majority of the change in orientation is caused by the movement of the platform. Therefore, we can use bundler to replace the orientation with it. This will work well when the orientation change is due to the platform only. This will model the case where the vehicle is driving along a straight road. However, it is not good in general.

World Coordinates

The previous two approaches are approximations. The fundamental reason why the appearance changes is because the vehicle moves through the environment, and it is viewed from different angles. Therefore, this set of features attempts to capture all these elements together. It consists of:

- $\mathbf{x}_t$
- time
- $(\mathbf{c}_t - \mathbf{x}_t)$ in polar coordinates, with radius in log scale.

4.5 Evaluation

To evaluate the performance of the prediction models we explored its ability to compute the joint track likelihood on a set of ground truth tracks. The intuition is as follows. If the GP can predict the measurement likelihood $L_{\mathbf{z},\overset{*}{\mathbf{x}}_k,\mathbf{e}}(\mathbf{x})$ accurately, it should produce a more accurate estimate.

Use ground truthed trajectories from the Four Hills dataset. The joint probability is

$$p(\mathbf{x}_{1:k}, \mathbf{z}_{1:k}) \;\propto\; p(\mathbf{x}_1)\prod_{i=1}^{k} L_{\mathbf{z},\overset{*}{\mathbf{x}}_i,\mathbf{e}}(\mathbf{x}_i)\prod_{i=2}^{k} p(\mathbf{x}_{i+1}|\mathbf{x}_i) \tag{4.6}$$

Conditioned on the ground truth dataset $\hat{\mathbf{x}}_{1:k}$, this simplifies to

$$p(\mathbf{z}_{1:k}|\hat{\mathbf{x}}_{1:k}), \;\propto\; \prod_{i=1}^{k} L_{\mathbf{z},\overset{*}{\mathbf{x}}_i,\mathbf{e}}(\hat{\mathbf{x}}_i) \tag{4.7}$$

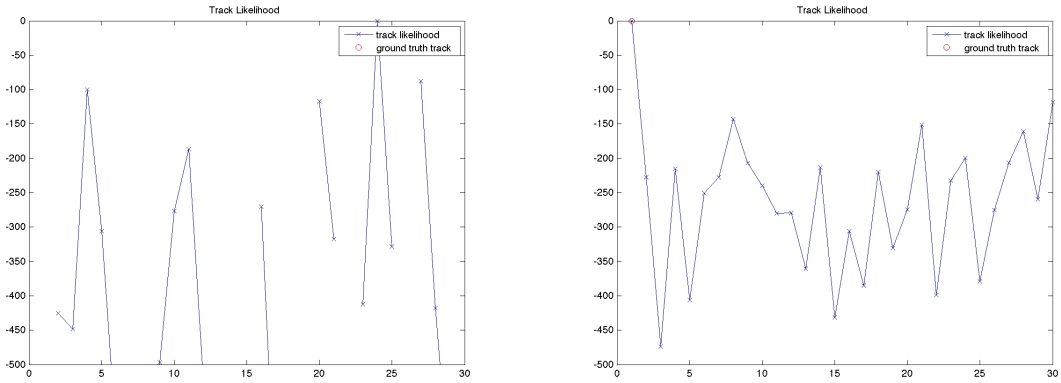
pseudolikelihood. As explained in Section 3.3.1, this term is a standard likelihood model which computes the probability of observing $\mathbf{z}$, conditioned on the pose of the platform, the state of the environment, and the fact that the target is in state $\mathbf{x}$. One way to assess the quality of this approximation is if this likelihood, when used in a maximum likelihood estimator, is able to correctly predict the correct state. In particular, consider a sequence of

Suppose a ground truth target $\hat{\mathbf{x}}$ together with an observation $\mathbf{z}$ are known. We should find that

$$L_{\mathbf{z},\overset{*}{\mathbf{x}},\mathbf{e}}(\hat{\mathbf{x}}) > L_{\mathbf{z},\overset{*}{\mathbf{x}},\mathbf{e}}(\hat{\mathbf{x}} + \delta\mathbf{x}), \tag{4.8}$$

where $\delta\mathbf{x}$ is a perturbation on the nominal state.

Figure 4.4: The observation associated with the ground truth and with the perturbed states.



(a) 90% of the variance is retained and the number of dimensions per feature is not limited.

(b) 90% of variance is retained, but the maximum number of dimensions per feature cannot exceed 10.

Figure 4.5: The log likelihood model using two constraints for PCA. The first ensures that the variance is 90% of the original for each feature and does not constrain the number of dimensions. Each graph show the relative log likelihood, scaled with respect to the maximum value of the likelihood function. The breaks in the lines for the unlimited case is caused by the fact that the likelihood becomes zero and the logarithm is undefined.

To test this, we explored the performance of several ground truthed tracks in the Four Hills training data. These provide a sequence of coordinates, $\hat{\mathbf{x}}$, together with observations. The perturbations $\delta\mathbf{x}$ were drawn from windows related with scale, rotation and offset. Figure 4.4 shows an example of the template associated with the ground truth observation, together with a set of peturbations.

Figure 4.5 plots the log likelihood results. As can be seen, both choices of the state space show a very strong return at the location of the vehicle, and very weak returns away from the vehicle. This shows they are discriminative. However, perhaps surprisingly, the performance of the 90% variance seems to produce slightly worse and numerically unstable results. This is likely to be caused by numerical issues and / or overfitting. Therefore, this initial investigation suggests that we will be

4.6 Clutter Model

In addition to modelling the likelihood that a given feature vector is generated by a target, it is also necessary to model the likelihood that a feature was generated by background clutter. Only by comparing these likelihoods, can we then estimate the probability that a given feature vector was actually generated by a target versus some other irrelevant feature in the environment.

Here, similar intuitions apply to appearance of clutter, as to targets. In particular, we may generally expect to see spatial and temporal correlations, since parts of the environment viewed close together in space and time are likely to appear similar. Moreover, different types of terrain are also likely to generate similar feature vectors. For example, one would expect different sections of road to look similar to each other, but different from buildings, vegetation, or other vehicles. With this in mind, we again model clutter as a Gaussian Process. However, in addition to the regression inputs described above, we also maintain separate means and covariance parameters for different types of terrain.[2] These models are then trained by

[2]Although terrain types may be user defined, and specified using maps and other prior knowledge, here we use an unsupervised approach, by automatically discovering terrain classes by clustering based on feature vector values.

computing feature vectors for parts of each camera frame at random, which are known not to be associated with a target.

Chapter 5

Implementation and Integration

5.1 Introduction

In this chapter, we outline our implementation of the PHD filter. This builds upon the PHD filter described in Chapter 3 and the observation model developed in Chapter 4. However, a crucial element is that the most effective results are obtained when we transform the tracking system into 3D. This required reformulating the tracking system.

The structure of this chapter is as follows. Section 5.2 provides an overview of how the tracking algorithm works. Section 5.3 describes how the 2D tracking problem was transformed into a full 3D problem.

5.2 Overview of the LoFT-PHD Implementation

The combined procedure for tracking each target that a user wishes to observe proceeds as follows.

1. At a time k_0, the operator indicates a vehicle they wish to be tracked. This target is labelled target i. The position of the target is computed in pixel coordinates, and a template $\mathbf{t}_{i,k_0}$ is extracted. A set of features $\mathbf{f}_{i,k_0}$ are extracted from this template and are used for subsequent matching. The PHD filter itself is initialised with a set of particles. The particles are clustered around the start location, and the sum of the weights of the particles is 1 because a single target has been detected.

2. Using an initial estimate of the target's heading and velocity, a search region is constructed around each particle's position, and is used to specify an ROI within the next camera frame, in which to search for the target at time $k+1$.

3. As described in the previous chapter, the ROI is decomposed into a set of n overlapping rectangular blocks, each representing a candidate successor position of the target's appearance in the camera's field of view at time $k+1$. Using the computer vision descriptors from LoFT, we then construct a feature vector, $\mathbf{f}_k^{[i]}$, for each block.

4. For all candidate $\mathbf{f}_k^{[i]}$ constructed above, we use our Gaussian Process observation model to compute the likelihood that each $\mathbf{f}_k^{[i]}$ was either generated by the target, or by background clutter. If for any given, $\mathbf{f}_k^{[i]}$, the ratio of these likelihoods passes a predefined threshold, its corresponding position is used to construct a *measurement*, $\mathbf{z}_{k+1,i}$ in Z_{k+1}. In this way, candidate positions that have low likelihood of being associated with the target are eliminated, leaving only a small number of measurements that have a high likelihood of being associated with the target position.

5. Using the measurement set, Z_{k+1}, constructed above, particles are updated in the usual way as described in Section 3.3.1. In particular, the history of all previous states of each particle now represents

an hypothesed track, which the target may have followed, and can be associated with the set of measured feature vectors used to update its state. Together, these feature vectors represent the target's changing appearance over time, if its position had followed a given particle. In principle, when updating a give particle's position in future timesteps, the history of all its associated feature vectors should be fed into the observation model, and used to compute the likelihood that any future measurement is associated with that particle. This is because a target's appearance in future timesteps is dependent on its appearance in previous timesteps. However, in the interest of computational efficiency, it may not be possible to maintain a history of all feature vectors associated with each particle. Instead, only the n most recent features associated with each particle may be maintained, since these are likely to have a higher correlation with a target's future appearance, thus leading to better predictions of its future position. Nevertheless, by maintaining different feature vectors for each particle, we no longer need to commit to a single template to estimate a target's current appearance. Instead, each particle is associated with a full probability distribution over a target's likely appearance given its previous measurements. In this way, we can maintain multiple hypotheses about a target's current position, and give a more complete picture of the uncertainty surrounding its current position.

In summary, we now have a complete procedure for maintaining multiple hypotheses about the current position of a single target, where each hypothesis corresponds to a given particle in the PHD filter implementation, and is weighted by the likelihood that it represents the target's true position. In addition, multiple targets can be handled in the same way, by simply allowing a user to initiate multiple tracks over time using the same procedure. Each target would then be associated with a different subset of particles within the PHD filter, but in all other ways, the procedure remains unchanged.

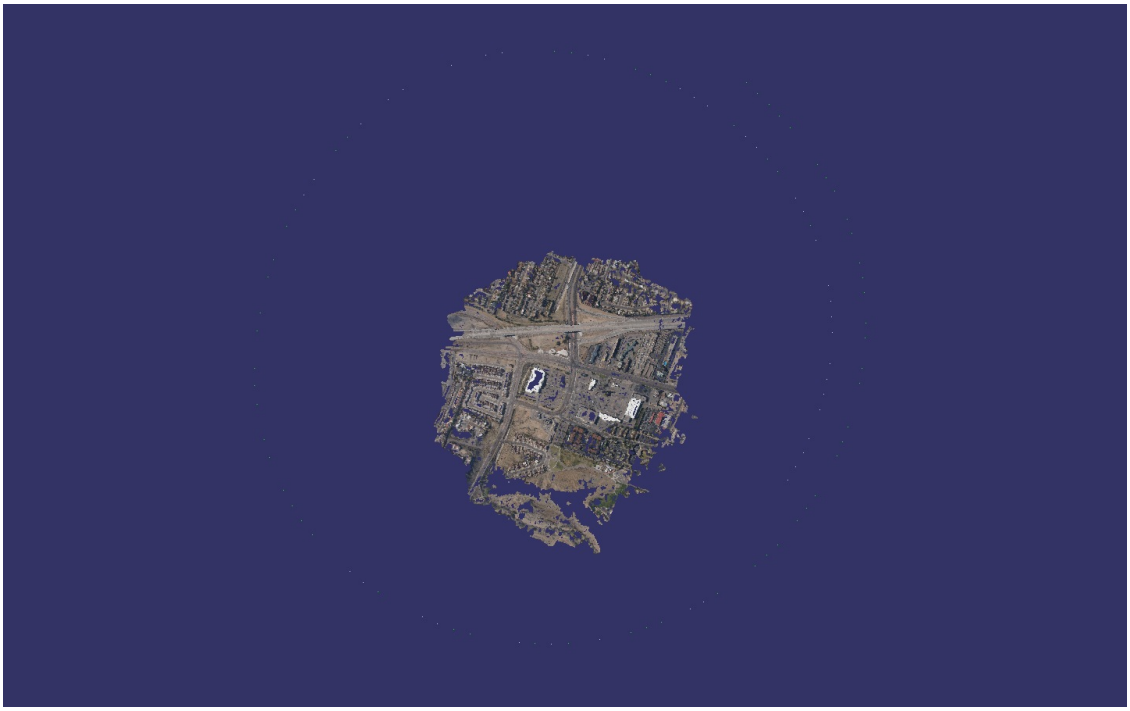
5.3 Transforming the Scenario into 3D

As explained in Chapter 4, appearance changes are fundamentally driven by the motion of the vehicle and the platform in 3D space. Therefore, rather than use the 2D tracking described in Section 2.3.2, we used a full 3D model. This involved two steps. First, a 3D model of camera motion and the enviornment was constructed using Bundler [11] and CMVS [3]. Figure 5.1 shows the bundled model which was constructed. Second, the cameras in the constructed model were aligned with the metadata on the location of the camera using a Procrustes minimisation algorithm. Figure 5.2 shows the camera metadata, together with the closest alignment of the cameras with the metadata. The covariance matrix of the errors in the alignments is given by

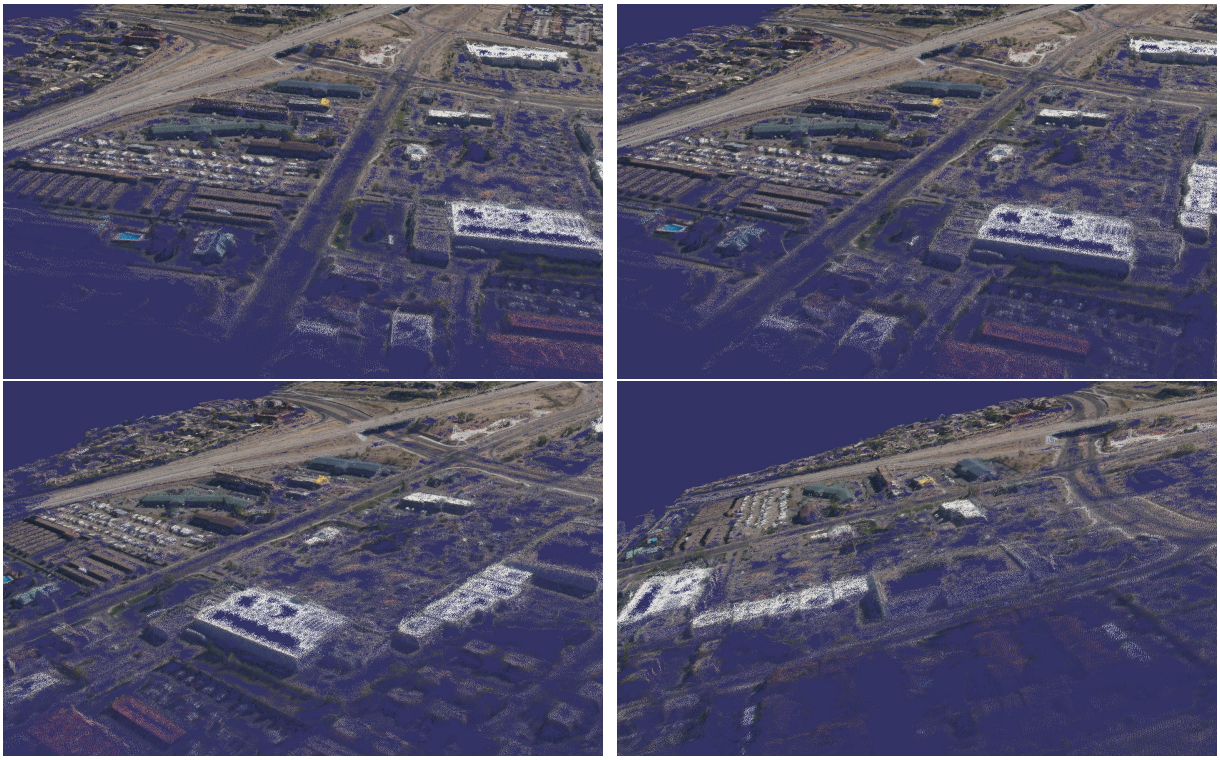
$$\begin{bmatrix} 21.1724 & 16.4079 & 0.3117 \\ 16.4079 & 28.4463 & 0.4906 \\ 0.3117 & 0.4906 & 0.6920 \end{bmatrix}. \tag{5.1}$$

Given the scale of the scenario, these errors are extremely small, and therefore we believe that a very accurate result has been obtained. This is confirmed by Figure 5.3, which overlays the detections used in the training data on the 3D model. To project these 2D projections into 3D space, the rays were intersected with a ground plane constructed from the

As can be seen, there is a good agreement between the detection locations and the road network. However, some misalignment can be seen. This is possibly due to slight angular errors.



(a) Overview of the bundled data. The small dots in a ring are the estimated camera poses, the dense model the reconstructed region.



(b) The dense 3D model produced using CMVS.

Figure 5.1: The 3D model constructed using Structure-from-Motion.

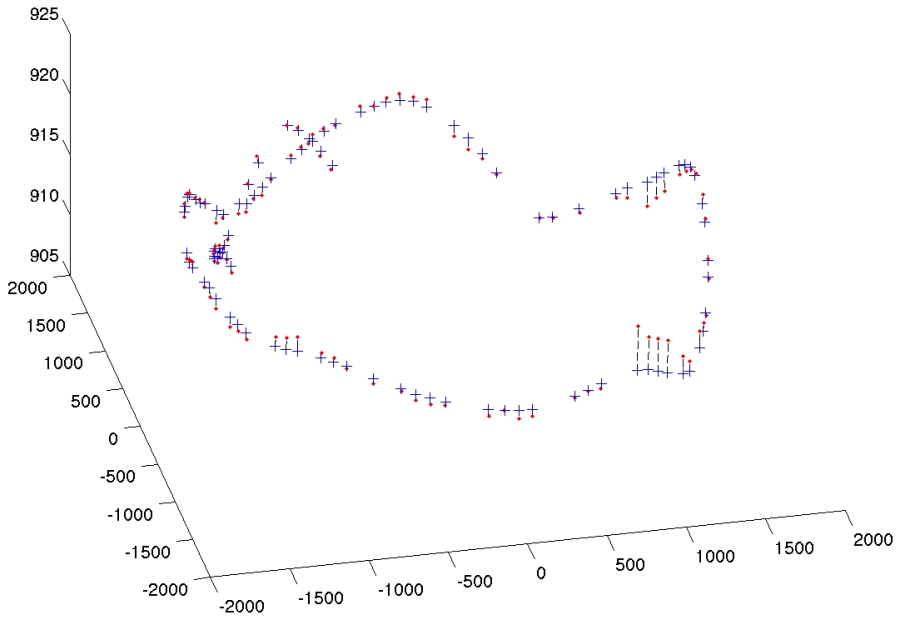


Figure 5.2: The metadata (blue crosses) and the aligned camera positions (red dots). Note that the horizontal and vertical scales are very different.

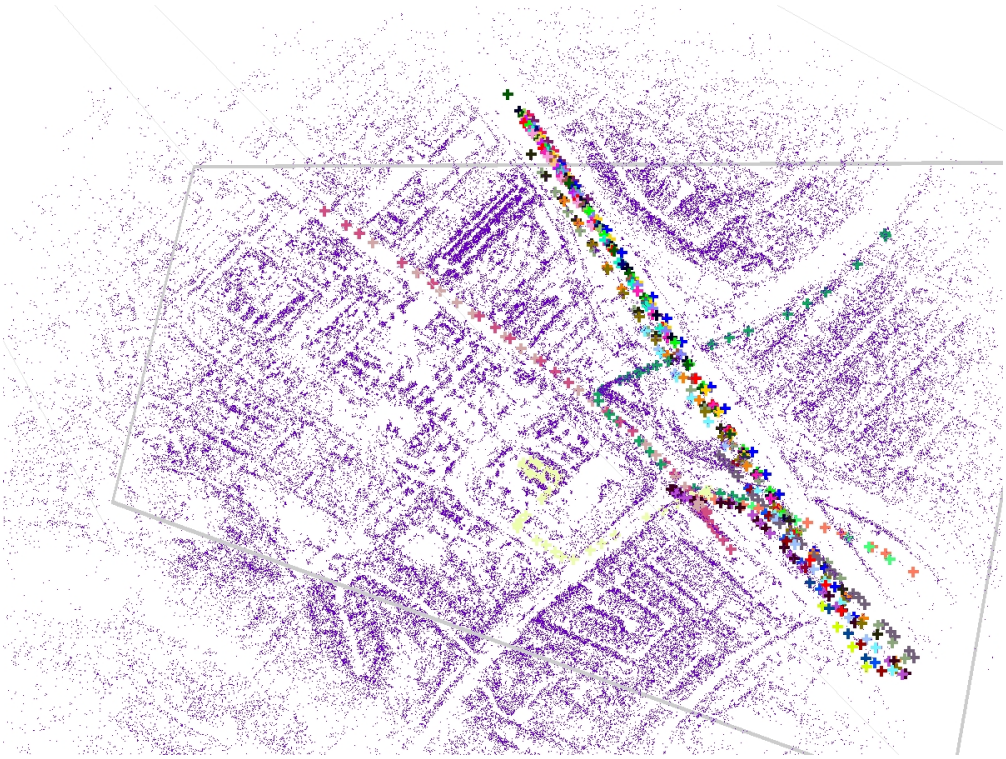


Figure 5.3: The projected detections from a ground truth dataset (crosses) projected on the point cloud (purple dots). The camera frustum is the grey trapezoid.

Chapter 6

Summary and Conclusions

6.1 Summary

This report has summarised the work to date on the project "Modelling and Characterisation of Detection Models in WAMI for Handling Negative Information". This project investigatesq how obervations models can be used in state of the art multi-target tracking algorithms. In particular, a machine learning technique known as Gaussian processes has been used to develop a model which explains how features extracted from a template describing vehicle appearance can evolve forwards through time.

6.2 Outstanding Work

Although this report describes work to date, there a number of issues which are currently in development.

6.2.1 3D Formulation of the Tracking Problem

As explained in Chapter 5, we are re-formulating the tracking problem natively in 3D. We have transformed the data into 3D, and we have

To be consistent with LoFT, we have posed the tracking problem purely in terms of pixel coordinates. However, this means it is not possible to properly model effects such as the change in distance between the vehicle and target, and the rotation caused by the orbit of the platform. Furthermore, uneven motions of the platform cause large, uncompensated movements in the locations of the targets, posing challenges to the trackers.

Therefore, the first thing we will do is to reformulate the tracking problem entirely in 3D. In particular, we will use the bundler-derived estimate of the extrinsic properties of the camera to model the motion of the camera through space. By doing this, we will be able to account for the relative attitude between the vehicle and the platform. The GP will be trained using the world coordinates as descrbed in Section 4.4.3. The motion models will be updated to describe the trajectory in 3D.

6.2.2 Training and Validation

We currently use 38 training tracks from the Four Hills dataset. We will seek to extend this, by generating further training sets within Four Hills, and also from other sets as well. We currently have access to the Albuquerque set.

Bibliography

[1] E. Blasch, P. Deignan, S. Dockstader, M. Pellechia, K. Palaniappan, and G. Seetharaman, "Contemporary concerns in geographical/geospatial information systems (GIS) processing," *IEEE National Aerospace and Electronics Conference (NAECON)*, pp. 183–190, 2011.

[2] M. R. Endsley, "Theoretical Underpinnings of Situation Awareness: A Critical Review," in *Situation Awareness Analysis and Measurement*, M. R. Endsley and D. J. Garland, Eds. Taylor & Francis, 2000, ch. 1, pp. 3–28.

[3] Y. Furukawa and J. Ponce, "Accurate, Dense, and Robust Multi-View Stereopsis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 8, pp. 1362–1376, August 2010.

[4] A. Haridas, R. Pelapur, J. Fraser, F. Bunyak, and K. Palaniappan, "Visualization of automated and manual trajectories in wide-area motion imagery," in *Int. Conf. Information Visualisation*, 2011, pp. 288–293.

[5] W. Koch, "On Exploiting 'Negative' Sensor Evidence for Target Tracking and Sensor Data Fusion," *International Journal of Information Fusion*, vol. 8, no. 1, pp. 28–39, January 2007.

[6] K. Palaniappan, R. Rao, and G. Seetharaman, "Wide-area persistent airborne video: Architecture and challenges," *Distributed Video Sensor Networks*, pp. 349–371, 2011.

[7] K. Palaniappan, F. Bunyak, P. Kumar, I. Ersoy, S. Jaeger, K. Ganguli, A. Haridas, J. Fraser, R. M. Rao, and G. Seetharaman, "Efficient Feature Extraction and Likelihood Fusion for Vehicle Tracking in Low Frame Rate Airborne Video," in *13th Conference on Information Fusion (FUSION)*, Edinburgh, UK, 26–29 July 2010, pp. 1–8.

[8] R. Pelapur, S. Candemir, F. Bunyak, M. Poostchi, G. Seetharaman, and K. Palaniappan, "Persistent Target Tracking Using Likelihood Fusion in Wide-Area and Full Motion Video Sequences," in *Proceedings of FUSION 2012*, Singapore, July 2012.

[9] R. Pelapur, K. Palaniappan, and Guna, "Robust Orientation and Appearance Adaptation for Wide-Area Large Format Video Object Tracking," in *9th International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*, Beijing, China, 18–21 September 2012, pp. 337–342.

[10] B. Ristic, D. Clark, B.-N. Vo, and B.-T. Vo, "Adaptive Target Birth Intensity for PHD and CPHD Filters," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, no. 2, pp. 1656–1668, April 2012.

[11] N. Snavely, S. M. Seitz, and R. Szeliski, "Photo Tourism: Exploring Image Collections in 3D," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2006)*, vol. 25, no. 3, pp. 835–846, July 2006.

[12] T. L. Song, D. Musicki, and K. D. Sol, "Target Tracking With Target State Dependent Detection," *IEEE Transactions on Signal*, vol. 59, no. 3, pp. 1063–1074, March 2011.

[13] R. Viguier, K. Palaniappan, E.Duflos, and P.Vanheeghe, "Particle Filter-Based Vehicle Tracking Using Fused Spatial Features and a Nonlinear Motion Model," in *Proceedings of SPIE*, 2012.