MULTIPLE MULTISTAGE HYPOTHESIS TESTS: A SEQUENTIAL DETECTION APPROACH TO TARGET TRACKING

by

HAYDN SCOTT RICHARDSON

A thesis submitted to the Department of Electrical
Engineering in conformity with the requirements
for the degree of Master of Science

Queen's University
Kingston, Ontario, Canada
June 1992

Abstract

A new algorithm, Multiple Multistage Hypothesis Test Tracking (MMHTT), is presented as a solution to an important class of multidimensional signal detection and estimation problems: the detection and tracking of multiple, low observable, moving point-source targets of unknown position and velocity, in a sequence of digital images. The MMHTT algorithm is a 'track-oriented' multiple hypothesis tracking (MHT) algorithm which exploits a new implementation of the recent multidimensional sequential detection algorithm, Multistage Hypothesis Testing (MSHT) to achieve joint target detection and tracking. The MMHTT algorithm is a maximum likelihood, multiple hypothesis, 'track-before-detect' algorithm which exploits the properties of sequential hypothesis testing to efficiently prune an exhaustive tree-structured search of candidate target trajectories. Detected trajectory segments are efficiently clustered in a localized track hypothesis data structure managed by a multiple hypothesis tracking scheme. Detection and tracking performance bounds for both the MMHTT and MSHT algorithms are derived and an implementation of the MMHTT algorithm is described including a new implementation of the MSHT algorithm. Finally, a system is presented for preprocessing video image sequences prior to applying the algorithm to the detection and tracking of object feature points in video image sequences.

Acknowledgements

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, there has been a growing interest in real-time image sequence analysis [1–4]. Recent advances in digital signal processing hardware and algorithms, and dramatic increases in the volume of image data being processed, are driving rapid growth in this field. Developing applications of this technology range from robotic vision systems to the automated analysis of dynamic imagery generated by spaceborne and airborne surveillance sensors.

Digital image sequence analysis is the field of study encompassing the techniques and algorithms for extracting dynamic information from sequences of digital images. Usually the goal of the analysis is to extract information from the image sequence which could not have been obtained from the static analysis of each individual image in the sequence. Each image in the sequence is the result of sampling and quantizing the output of an electro-optical sensor, such as a video or infrared CCD (charge-coupled device) array or imaging radar. In general, the sensor may be either fixed or moving with respect to the imaged scene, but appropriate processing can often compensate for the effects of sensor motion.

Advances in electro-optical sensor technology have enabled the development of wide-field-of-view surveillance applications with a concomitant interest in the development of algorithms to detect and track small, low observable targets in remotely sensed imagery. However, the successful development of these algorithms is dependent on the solution of fundamental problems in multidimensional signal detection and estimation. Although the detection and tracking of low observable targets has been an active area of research in the radar and sonar communities for nearly thirty years [5–20], the detection and tracking of low observable targets in optical images has been the subject of intense investigation for little more than a decade [21–43].

The use of high speed digital signal processors and higher quality optics has greatly affected the design and implementation of these applications. Increasingly demanding system requirements and a steady decrease in the cost of high performance processors, have made highly sensitive but computationally intensive algorithms more attractive for real-time applications. Current research and algorithm development is focused on improving detection and tracking performance and efficiency, particularly for low observable targets in moderate to dense target environments.

The problem of low observable point-source target detection and tracking arises in remote surveillance applications where the targets are sufficiently distant from the sensor that their focal plane image is determined by the sensor's optical point-spread function (psf) ($\sim 1-20$ pixels) and the target signal amplitude is weak relative to the background clutter and noise recorded by the image sensor. The small spatial extent of these targets precludes the use of many traditional image processing techniques which exploit information about a target's size, shape and features [44–46]. In fact, the problems associated with small targets were considered of sufficient interest that the Society of Photo-Optical Instrumentation Engineers (SPIE) recently inaugurated an annual conference on the Signal and Data Processing of Small Targets [47–49].

SPIE defines small targets as having a total spatial extent of less than 80 pixels (9x9) which is less than .15% of a 256x256 pixel image. This classification includes

- point-source targets
- small extended objects, and
- clusters of point-source targets and small extended objects [48].

1.1 Applications

The following applications are representative of the environments in which small target detection and tracking plays a critical role:

Wide-Field-of-View Telescopes

Recently, there has been significant interest in the development of Wide-Field-of-View Telescopes to locate, track and catalog the increasing number of artificial satellites orbiting the Earth [22, 29, 33, 40, 42, 50, 51]. An array of CCD detectors on the focal surface of the telescope, integrates the photon charges received during a frame sampling period to construct a digital image of the telescope's field-of-view (FOV). Typically, long exposure times are used to detect faint objects but this technique is only effective for stationary objects. The system frame rate for moving objects requires that each image be constructed from a relatively small number of photons. Consequently, the resultant imagery is dominated by random sensor noise. The problem, in this case, is to detect and track meteors, artificial satellites and other small, luminous targets in the resulting digital telescope image sequences of the night sky. Due to the immense distances involved, the targets can be approximated as point-source targets in the telescope imagery.

Infrared Search and Track Systems

It is becoming common for passive electro-optical (EO) sensors to be included in the design of airborne and terrestrial military platforms. Passive EO sensors are valued for their lack of sensor emissions, electronic countermeasures (ECM) immunity, and their ability to provide improved recognition and verification through the presentation of recognizable images [52]. Infrared sensors have the additional advantages of night operability and improved haze penetration.

Infrared Search and Track (IRST) systems are wide-field-of-view surveillance systems designed for autonomous target detection and track acquisition [53]. An IRST system typically uses a Forward Looking Infra Red (FLIR) sensor to detect moving air and ground targets. Since maximum range detection is critical, IRST systems are essentially point-source detectors. The targets are typically buried in highly structured background clutter and have a very low signal-to-noise-ratio. Detection is often hampered by sensor and background motion which degrades the performance of clutter and noise suppression filters, and by the absence of *a priori* target and background signature information.

Space-Based Surveillance and Tracking Systems

There has been considerable interest over the last decade in developing Space Based Surveillance and Tracking Systems (SSTS) to simultaneously detect and track a large number of fast-moving targets, such as airplanes or missiles, over a wide-field-of-view. A satellite-borne staring or scanning mosaic sensor staring at a fixed point on the ground can be designed to detect and track a particular class of targets [26, 54]. Maximizing the achievable performance under computational resource constraints is of major concern for aerospace surveillance applications as considerations of on-board processor size, weight and power consumption restrict the computational complexity of systems designed for autonomous aerospace-borne sensors.

Video Tracking

Real-time video tracking is of considerable interest in automating the analysis of film and video. Video analysis is used in applications ranging from videotheodolites for motion analysis of military test range imagery [46], to commercial systems to analyze a golf swing or baseball pitch. Video image sequences typically contain highly structured, non-stationary background clutter with a good signal-to-noise ratio. However, video tracking algorithms must be able to effectively cope with a rapidly time-varying background, target occlusion and/or sudden changes in local or global illumination.

Three-Dimensional Rigid-Body Motion Estimation

Researchers in computational vision have been studying the problem of estimating three-dimensional motion from a sequence of images, intensively [1, 3, 4, 55]. Although the objects of interest are typically quite large (greater than 80 pixels), there is an extensive body of research in three-dimensional rigid-body motion estimation [55–58] which assumes the prior existence of a set of feature point correspondences. These algorithms can compute the three-dimensional motion of a rigid-body given the location of known feature points in each image of a sequence. The detection and tracking of these feature points is suitable for consideration within the small target detection and tracking framework [59, 60].

1.2 The Target Tracking Problem

The basic problem inherent to the previous applications is the detection of small, low observable, moving targets in a sequence of digital images and subsequent estimation of the target trajectories. Hereinafter, this problem will be referred to as the Multiframe Target Detection and Tracking (MFTDT) problem. In each case, an image sensor produces an image sequence which may contain one or more moving targets. Each image in the sequence is a function of the observed scene, the sensor optics, and the system hardware.

Figure 1.1: The Three-Dimensional Image Sequence Volume

The analog signal received by the sensor hardware is sampled in both space and time, and quantized to a discrete set of image pixel intensities. The resulting digital image sequence is a triply indexed, discrete volume of digital image intensity data $I[x, y, t]$ (see figure 1.1), where $\mathbf{x} \equiv [x, y]$ are the spatial coordinates of discrete image pixel locations and t is the temporal frame index. In general, $I[\mathbf{x}, t]$ may be multi-valued representing, for example, the received intensity in different spectral bands of a multispectral sensor.

When a low observable point-source target is present in the sensor's field-of-view, the resulting image sequence will be a function of both the background scene and the target. The problem is to detect the target and track its trajectory through the image volume. The difficulties of this task are quite clear. The small spatial extent of the target limits the information content of the target signature precluding the use of traditional pattern matching approaches, and the signal-to-noise ratio is sufficiently low that detection specifications cannot be met by an analysis of a single image frame.

This suggests that the target signal energy should be integrated along its trajectory. However, the position and velocity of the target are initially unknown and the target may follow an arbitrary trajectory. In many applications, multiple targets are present in the same sequence and the image sequence may be contaminated by non-stationary background clutter and varying amounts of random noise.

Traditionally, detection and tracking have been treated as two separate problems. The standard approach has been to filter the sensor data, in an attempt to suppress the background clutter and noise, and then to detect targets by applying a fixed or adaptive threshold to each image. With this approach, every image pixel which exceeds the detection threshold is declared a target observation. The tracking algorithm then attempts to associate detection observations with target tracks governed by a state-space model of target kinematics and observations.

Figure 1.2: Basic System Structure for Detecting and Tracking Small Targets

The basic algorithmic structure for the detection and tracking of small targets is outlined in figure 1.2. A distinction is made between signal processing functions which operate on image data and data processing functions which operate on higher-level data abstractions such as target state estimates and data association hypotheses. Typically, the signal processing functions must rapidly process large volumes of image data, making accurate, reliable detection decisions while the data processing functions partition the detected target observations into target tracks, estimate target state parameters and predict future target behaviour.

There are three major functional groups to this processing structure:

1. image preprocessing, including
   (a) corrections for known sensor distortions,
   (b) image registration to correct spatial alignment differences between images,
   (c) and clutter suppression filters to enhance the signal-to-noise ratio.
2. signal detection and estimation, including
   (a) target detection,
   (b) and estimation of target position, velocity and signal strength
3. and multitarget tracking, including.
   (a) state estimation,
   (b) data association,
   (c) and the generation and evaluation of target track hypotheses.

There is an apparent ambiguity in the division of the signal and data processing functions in figure 1.2. This grey area between functional blocks is due to the merging of target detection and tracking functions in emerging multiframe 'track-before-detect' algorithms. Classically, target detection algorithms produced *hard* (target present/target absent) detection observations for each frame and then the tracking algorithm made *observation-to-track* association decisions. These 'detect-before-track' algorithms can be computationally simple, but they only exhibit reasonable performance when the signal-to-noise ratio is high.

It is increasingly recognized that in order to provide a detection and tracking capability for small targets under more adverse conditions, processing procedures must make optimal use of all the available information at every processing stage. This implies that *hard* detection decisions should be avoided or delayed. Recently, an interest in improving detection performance for low observables has led to the development of so called 'track-before-detect' algorithms which make tentative data association decisions over several image frames prior to target detection. These detection algorithms achieve enhanced detection performance by incorporating low level data association functions in the detection process. The detection/tracking problem is then one of optimally processing the data so as to meet detection and tracking specifications while maintaining a practical computational load.

The major problem confronting system designers, when selecting an appropriate multitarget tracking (MTT) algorithm for a given target environment, is the trade-off between the tracking performance achieved and the computational resources (processor instructions/sec, data memory, etc.) required for its implementation. In aerospace applications, where size, weight and power consumption are critical attributes, the computational resources available to an MTT application are firmly constrained. Thus, the computational resources required to meet the minimum system requirements for target detection and tracking, and the maximum achievable performance given the available computational resources, are issues requiring careful consideration.

Consequently, the primary measure of performance for a given algorithm, is its computational efficiency in achieving a given level of performance. Accurate predictions of an algorithm's performance for realistic target environments usually involve complex models of the target environment and extensive simulations. However, simpler models can be used to analyze the relative performance trade-offs involved in allocating scarce computational resources.

In the following, we address the detection and tracking problems jointly. Our approach is to combine the concepts of 'track-before-detect' [22, 25, 26, 32, 33, 43], sequential detection [40] and multiple hypothesis tracking [7, 35]. This research can be viewed as an extension of recent developments in image processing [22, 25, 33, 39, 40], target tracking [26, 28, 35] and computer vision [59, 60]. The new algorithm combines the strengths of sequential signal detection and Bayesian multiple hypothesis tracking. The basic objectives of this research were to determine the feasibility of this approach, and to provide explicit means for analyzing the detection and tracking performance.

1.3 Overview

This document will describe the design and development of the Multiple Multistage Hypothesis Test Tracking algorithm and its application to the establishment of feature point correspondences. Chapter 2 discusses relevant prior research in target detection and tracking and introduces a point-source target model for image sequences. The MMHTT algorithm is developed in Chapter 3 and is illustrated with a simple example of two intersecting target trajectories. Chapter 4 outlines techniques for evaluating the algorithm's detection and tracking performance for arbitrary known target scenarios. Finally, in Chapter 5, a system for automating the establishment of feature point correspondences, for the estimation of three-dimensional structure and motion, is presented as an application of the MMHTT algorithm.

1.4 Thesis Contributions

The main contributions of this thesis are:

1. A new implementation of the Multi-Stage Hypothesis Test algorithm.

The key difference between the implementation of the MSHT algorithm in [40] and the new implementation presented herein is the method of managing candidate target trajectories which have been neither confirmed nor rejected. In [40], these trajectories were stored relative to the pixel in which they originated. In the new implementation, these trajectories are stored relative to their location in the most recent image (see section 3.1). Thus, there is a list of current candidate trajectories associated with each image pixel location.

This local indexing of the undecided trajectories has two important consequences. First, it enables an evaluation of the current candidate trajectories at a pixel to influence the initiation of a search for new targets originating in that pixel. In [40], a search for new targets was initiated for every pixel in every image frame. In section 3.3.1, a decision feedback process is introduced which suppresses the initiation of a new target search where the current pixel is more likely to contain an observation of a previously detected target than an observation of a new target. The result is an improvement in the computational efficiency of the MSHT algorithm. The second consequence of the local indexing of undecided trajectories is an ability to incorporate the MSHT algorithm in a multiple hypothesis tracking scheme.

2. A sequential algorithm for joint target detection and tracking.

Multiple hypothesis tracking (MHT) [7] is a term which refers to a class of data association algorithms that are noted both for their excellent error performance and for their formidable computational requirements. MHT algorithms generate an exhaustive set of data association hypotheses to describe the detected target observations. These algorithms can be characterized by two main elements: 1) a method for managing the multiplicity of generated hypotheses and 2) a ranking metric to evaluate competing hypotheses.

The MSHT algorithm is an efficient sequential detection algorithm which detects short, linear segments of a target's trajectory but does not have an explicit mechanism to extend the detected trajectory segments. It is shown in section 3.2.1, that the likelihood-ratio test statistic used by the MSHT algorithm can be used as a ranking metric for track hypotheses. In sections 3.2.3–3.2.5, this statistic is used to develop a new multiple hypothesis tracking algorithm, Multiple Multistage Hypothesis Test Tracking (MMHTT), which employs repeated MSHTs for combined target detection and tracking.

The development of this new, 'track-oriented' MHT algorithm follows the development of the Structured-Branching MHT (SB-MHT) algorithm in [35]. Although both algorithms employ a sequential probability ratio test in their evaluation of competing track hypotheses, the two algorithms are distinct. The SB-MHT algorithm employs a likelihood function derived from a state space model of the expected target dynamics while the MMHTT algorithm unifies the target detection and tracking problems with a likelihood function derived from a model of the sensor signal in the presence/absence of a target. The result is a new sequential algorithm for *joint* target detection and tracking which combines the detection performance and efficiency of the MSHT algorithm with the well-documented tracking performance of a multiple hypothesis tracking approach [5, 7, 35].

3. An analysis of the performance of the MSHT and MMHTT algorithms.

An ability to predict and analyze the performance of a multitarget tracking algorithm is invaluable when designing a MTT system. The performance analysis of the MSHT algorithm in [40] was restricted to an analysis of single candidate trajectories with independent and identically distributed observations. This analysis has been generalized, in section 4.1, to evaluate the performance of multistage hypothesis tests with independent but non-identically distributed observations. In addition, the analysis of the MSHT algorithm has been extended, in sections 4.2 and 4.3, to include an analysis of the probability of false alarm and the probability of missed detection for full candidate trajectory sets and for the repeated MSHT which characterizes the MMHTT algorithm. The tracking performance of the MMHTT algorithm has also been analyzed and explicit procedures for estimating the average time to track loss has been derived in section 4.3.1.

4. An application of the MMHTT algorithm to feature detection and tracking in video image sequences.

The establishment of feature point correspondences over an extended number of image frames is a critical requirement of many algorithms for the estimation of three-dimensional structure and motion. Feature detection and correspondence were addressed in [60] by applying the MSHT algorithm to detect image features along short, linear trajectories through several images. Although this multiframe detection approach offered increased robustness to feature detection problems and provided implicit local correspondence decisions, it failed to extend the detected feature trajectories. Consequently, a heuristic feature path linking algorithm was proposed to produce the long trajectories required for multiframe estimation of structure and motion. In Chapter 5, the MMHTT algorithm is proposed as an alternative solution to feature detection and tracking which obviates the need for the heuristic path linking step.

Chapter 2

Small Target Detection and Tracking

This chapter outlines the major existing approaches to the detection and tracking of small, moving targets in digital image sequences. In the following, algorithms for the signal and data processing of small targets will be discussed in terms of their underlying assumptions about the image sequence data and their ability to make optimal use of the available information. Each algorithm finds application under a suitable set of conditions but selection of the most appropriate algorithm for a given problem is a non-trivial task. Ideally, algorithms should be compared on their ability to efficiently meet system detection and tracking performance requirements. However, it is often difficult to compare the expected performance of two candidate algorithms without performing extensive simulations.

The chapter begins with the development of an image signal model for point-source targets, a review of signal detection and a discussion of multidimensional signal detection algorithms, including single and multiframe approaches. Then the fundamental concepts of multitarget tracking are reviewed and the major existing approaches described. The convergence of signal and data processing algorithms in emerging 'track-before-detect' techniques is presented as a response to increasingly demanding system performance requirements culminating in the recent development of Bayesian Multiple Hypothesis Techniques for joint target detection and tracking. The chapter concludes with a discussion of the relationship between the MFTDT problem and the feature correspondence problem in computer vision.

2.1 The Point-Source Target Model

In this section, a signal model will be developed for the discrete image data $I[x, y, t]$ in the MFTDT problem. As discussed in Chapter 1, the point-source target detection problem arises in remote surveillance applications where the targets are sufficiently distant from the sensor that their focal plane image is determined by the sensor's optical point-spread function. With typical electro-optical image sensors, the target's focal plane image is detected by an array of photodetectors on the sensor's focal plane.

In most remote imaging applications, the background illumination is incoherent and the optical image $I_i(x, y, t)$ incident on the detector array can be expressed as the spatial convolution of the sensor's optical point-spread function $p(x, y)$ with the apparent radiance of the imaged scene $I_r(x, y, t)$ [61, 30].

$$I_i(x, y, t) = I_r(x, y, t) * p(x, y) \tag{2.1}$$

Although the point-spread function may be spatially-varying in general, it will be assumed in the following that $p(x, y)$ is space invariant.

The apparent scene radiance $I_r(x, y, t)$ is a combination of the optical energy radiating from the image background and any targets which may be present. If the targets are transparent to the background illumination or sufficiently small relative to the resolution of the image sensor, then they may be considered additive to the background scene $I_b(x, y, t)$ and the apparent scene radiance can be modelled as

$$I_r(x, y, t) = I_b(x, y, t) + A(t)\,\delta(x - x_t(t), y - y_t(t)) \tag{2.2}$$

where each moving point-source target is modelled as a spatial impulse function

$$A(t)\,\delta(x - x_t(t), y - y_t(t)) \tag{2.3}$$

with position $(x_t(t), y_t(t))$ and signal amplitude $A(t)$ at time t. If the target obscures the background intensity, then the apparent scene radiance can be expressed as

$$I_r(x, y, t) = I_b(x, y, t) + (A(t) - \gamma I_b(x, y, t))\,\delta(x - x_t(t), y - y_t(t)) \tag{2.4}$$

where γ expresses the percentage of the background intensity occluded by the target.

The discrete image data $I[x, y, t]$ is obtained by sampling and quantizing the optical radiation incident on the sensor focal plane. A general model for this image formation process is

$$I[x, y, t] = q(d(I_i(x, y, t)) + n_{SN}(d(I_i(x, y, t))) + n_{WB}(x, y)) \qquad (2.5)$$

where

- $q()$ is the sensor quantization function,
- $d()$ is the photodetector response function,
- $n_{SN}()$ is a signal dependent shot noise process (Poisson pdf), and
- $n_{WB}()$ is a signal independent wide-band noise process (Gaussian pdf).

For notational convenience, $q()$ will be omitted in the following, but its presence will be implied.

In general, the detector response is nonlinear, but if the detector is operating in its linear region (2.5) can be approximated by

$$I[x, y, t] = I_i(x, y, t) * d(x, y, t) + \mu_{I_i} n_{SN}(x, y) + n_{WB}(x, y) \qquad (2.6)$$

where μ_{I_i} is the average intensity on the sensor focal plane. For example, a reasonable model for the image data produced by an ideal staring mosaic array of CCD detectors is

$$I[x_j, y_k, t_l] = \mathcal{R}_V^p \int_{(l-1)\Delta t}^{l\Delta t} \int_{(k-1)\Delta y}^{k\Delta y} \int_{(j-1)\Delta x}^{j\Delta x} I_i(x, y, t)\, dx\, dy\, dt + N[j, k, l] \qquad (2.7)$$

where $\mathcal{R}_V^p$ is the responsivity of the detector ($V\, q^{-1}\, sec^{-1}\, cm^{-2}$), $N[j, k, l]$ is the detector noise process, and it is assumed that the detector pixels are spatially contiguous rectangles with dimensions Δx by Δy and integration period (exposure) Δt [62, 33]. In the following, it will be assumed, without loss of generality, that the detector elements are spatially contiguous rectangles with uniform response and that the sensor image data can be modelled as in (2.7).

Substituting (2.4) into (2.1) and then into (2.7) yields the following expression for the image pixel data

$$\begin{aligned} I[x_j, y_k, t_l] &= \textstyle\int\!\!\int\!\!\int I_b(x,y,t) * p(x,y) - \gamma I_b(x_t(t), y_t(t)) p(x - x_t(t), y - y_t(t)) dx\, dy\, dt \\ &\quad + \textstyle\int\!\!\int\!\!\int A(t)\, p(x - x_t(t), y - y_t(t)) dx\, dy\, dt \\ &\quad + N[j,k,l] \end{aligned} \tag{2.8}$$

where $\gamma = 0$ when the target is additive to the image background, and $\gamma = 1$ when the target completely occludes the image background. Thus the distribution of optical intensity on the surface of the photodetector can be modelled as two separate components, a background clutter component and a target component.

For an ideal imaging system ($p(x,y) = \delta(x,y)$) with a uniform detector response and an infinitessimal exposure, the target impulse response is preserved and a point-source target is imaged as a point on the sensor focal plane. Consequently, the target photons are focused on a single detector pixel and the target signal contributes to exactly one pixel in the resulting image frame. Thus, the detector response is independent of the target's subpixel location on the detector surface.

However, any imaging system has a finite spatial resolution and the target photons may be distributed across several detector pixels depending on the sensor's point-spread function and the target's motion. Given the target trajectory in continuous-valued focal plane coordinates, the relative target intensity received by each pixel can be modelled as a fraction of the total target photons received by the sensor array during a single integration period [33].

$$w_{j,k,l}(x_t(t), y_t(t)) = \frac{\int_{(l-1)\Delta t}^{l\Delta t} \int_{(k-1)\Delta y}^{k\Delta y} \int_{(j-1)\Delta x}^{j\Delta x} p(x - x_t(t), y - y_t(t))\, dx\, dy\, dt}{\int_{(k-1)\Delta t}^{k\Delta t} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x - x_t(t), y - y_t(t))\, dx\, dy\, dt} \tag{2.9}$$

If the total target intensity incident on the focal plane during a single frame exposure is defined as

$$A = \int_{(k-1)\Delta t}^{k\Delta t} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} p(x - x_t(t), y - y_t(t))\, dx\, dy\, dt \tag{2.10}$$

then the image of a point-source target with focal plane coordinates $(x_t(t), y_t(t))$ can be expressed as $Aw_{j,k,l}(x_t(t), y_t(t))$.

For sensors with uniform detectors and space-invariant point-spread functions, the target image can be expressed as a function of the target's subpixel focal plane coordinates relative to the nearest sampling point of the detector array (x_s, y_s).

$$\begin{aligned} \mathbf{t} &= Aw_{j,k,l}(x_t(t) - x_s, y_t(t) - y_s) \\ & x_s - \tfrac{\Delta x}{2} \leq x_t(t) \leq x_s + \tfrac{\Delta x}{2}, \\ & y_s - \tfrac{\Delta y}{2} \leq y_t(t) \leq y_s + \tfrac{\Delta y}{2}, \\ & (l-1)\Delta t \leq t \leq l\Delta t \end{aligned} \tag{2.11}$$

In principle, $w_{j,k,l}(x_t(t)-x_s, y_t(t)-y_s)$ is a continuous function of $(x_t(t)-x_s, y_t(t)-y_s)$.

In the following, it will be assumed that

1. the imaging system can be modelled as a space-invariant linear system,

2. the target intensity can be modelled as additive to the background clutter,

3. the target intensity is constant,

4. the target trajectory can be approximated by a linear, constant velocity trajectory over a short time interval, and

5. the detector response is uniform and can be modelled as in (2.7).

2.2 Multidimensional Signal Detection

In this section, basic signal detection concepts will be discussed and applied to the MFTDT problem. The image model developed in section 2.1 will be used to define a binary hypothesis test for the presence or absence of a point-source target. This model will then be used as a reference to discuss previous approaches to target detection for the MFTDT problem. The simple problem of detecting an ideally imaged, constant intensity point-source target in an additive independent, identically distributed Gaussian noise background will be developed as an illustration of the various approaches.

It is generally accepted that the image formation process is fundamentally a statistical phenomenon [61]. The random fluctuations in detected energy are a function of discrete quantum interactions between light and matter which cannot in principle be perfectly predicted. This uncertainty leads to the development of probabilistic models for the received signal observations and places signal detection in the general framework of statistical inference [63, 64, 65].

The goal of target detection is to decide, based on a set of received signal observations $\mathbf{y}$, whether a target is present in the image sequence. The target detection problem can be viewed as a binary hypothesis testing problem with hypotheses

- H_1: Target Present and
- H_0: Target Absent.

Given a set of received observations $\mathbf{y} \equiv \{y_1, \dots y_n\}$, which can be modelled as realizations of random variables $\mathbf{Y} \equiv \{Y_1, \dots Y_n\}$ with joint probability distribution function (pdf) $f_{\mathbf{Y}}(\mathbf{y} \mid \theta)$ these hypotheses can be modelled as

$$\begin{aligned} H_1 : \mathbf{Y} &\sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta) \quad \text{with } \theta \in \theta_{H_1} \text{ and} \\ H_0 : \mathbf{Y} &\sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta) \quad \text{with } \theta \in \theta_{H_0} \end{aligned} \tag{2.12}$$

where Θ is the set of all possible values of the parameter vector θ, θ_{H_1} and θ_{H_0} are disjoint subsets of Θ such that $\Theta \equiv \theta_{H_1} \cup \theta_{H_0}$ and $\theta_{H_1} \cap \theta_{H_0} = \emptyset$. The problem is to design a decision rule $\delta(\mathbf{y})$ which partitions the set of realizable observations $\{\mathbf{y}\}$ into two regions Γ_0 and Γ_1, such that observations in Γ_i correspond with a decision to accept hypothesis H_i.

For the binary hypothesis testing problem, there are two types of decision errors.

1. Type I: Accepting H_1 when H_0 is true (False Alarm), and
2. Type II: Accepting H_0 when H_1 is true. (Missed Detection).

We will define the probability of a Type I error (probability of false alarm, P_{FA}) as α and the probability of a Type II error (probability of missed detection, $1 - P_D$) as β. These error probabilities are an important measure of the performance of a given decision procedure and are often quoted in system performance requirements.

The performance of a given decision rule can be expressed as a function of the signal parameter vector θ. A plot of the probability of detection versus θ, the *power function*, is one measure of the detection performance of a decision rule under different signal models. A more complete performance measure is the family of plots of the probability of detection versus the probability of false alarm, indexed by θ. This plot is commonly referred to as the *Receiver Operating Characteristic* (ROC) [63].

The performance of a given decision procedure is dependent on its ability to *separate* the mapping of received observations under hypothesis H_1 (target signal) from the mapping of received observations under hypothesis H_0 (background signal). In turn, the separability of the target and background signals is dependent on their relative statistical dependence and signal strengths. For some signal models, detection performance can be expressed as a function of the *distance* between the signal distributions of (2.12).

Given an ability to analyze the performance of a given decision rule, the question arises as to how to design an *optimal* decision rule. Ideally, an optimal decision rule would both minimize the probability of false alarm, α, and maximize the probability of detection, $1-\beta$. However, these are conflicting requirements. Various *cost functions* (Bayes, minimax and Neyman-Pearson) have been designed to balance the relative importance of these two types of decision error [63]. These cost functions lead to decision rules which can be described as the comparison of a likelihood ratio function to a decision threshold, τ

$$\delta(\mathbf{y}) = \begin{cases} 1, & > \tau \\ r, \quad \frac{f_{\mathbf{Y}}(\mathbf{y}|\theta \in \theta_{H_1})}{f_{\mathbf{Y}}(\mathbf{y}|\theta \in \theta_{H_0})} & = \tau \\ 0, & < \tau \end{cases} \tag{2.13}$$

where $\delta(\mathbf{y})$ is the probability of accepting hypothesis H_1.

In practice, the most common signal detection problem is the detection of a known signal in additive noise [63, 64]. This is a binary hypothesis testing problem where the observations $\mathbf{y}$ can be expressed as

$$\begin{aligned} \mathbf{y} &= \mathbf{s_1} + \mathbf{n} \text{ under } H_1 \text{ and} \\ \mathbf{y} &= \mathbf{s_0} + \mathbf{n} \text{ under } H_0, \end{aligned} \tag{2.14}$$

The signals $\mathbf{s}_i$ for this problem may be deterministic and completely known, deterministic with unknown parameters (e.g. amplitude) or they may be non-deterministic signals with known probability density functions. The detection algorithms which follow are all variations of this basic problem.

If the observations y_i are realizations of mutually independent and identically distributed random variables Y_i, then evaluation of the decision rule is greatly simplified. Defining a likelihood ratio function $L(y_i)$ as

$$L(y_i) \equiv \frac{f(y_i \mid \theta \in \theta_{H_1})}{f(y_i \mid \theta \in \theta_{H_0})} \tag{2.15}$$

the decision rule (2.13) can be expressed as a function of $\prod_{i=1}^{n} L(y_i)$. Since $\ln x$ is a strictly increasing function of x, (2.13) can be expressed as a log-likelihood decision rule

$$\delta(\mathbf{y}) = \begin{cases} 1, & > \ln \tau \\ r, \quad \sum_{i=1}^{n} \ln L(y_i) & = \ln \tau \\ 0, & < \ln \tau \end{cases} \tag{2.16}$$

This decision rule suggests a simple, easily analyzable detector structure consisting of a time-varying point non-linearity followed by an accumulator and a threshold comparator.

In general, the performance of any optimal decision procedure will improve as the number of observations increases. Many of the algorithms that follow attempt to exploit this property. However, the decision procedures for the binary hypothesis testing problem assume that either

$$\begin{aligned} \mathbf{Y} \sim \quad & f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \theta_{H_1}) \text{ or} \\ \mathbf{Y} \sim \quad & f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \theta_{H_0}) \end{aligned} \tag{2.17}$$

If the binary hypothesis model is compromised and the received observations are not identically distributed, the detection error rate will increase.

2.2.1 Point-Source Target Detection

Given a model for the image formation process, the detection of a point-source target with a known trajectory can be treated as the detection of a known signal in additive noise. The observation set $\mathbf{y}$ for this problem consists of the set of image pixels

$$\mathbf{y} = \{I[x_j, y_k, t_l] \mid w_{j,k,l}(x_t(t), y_t(t)) \neq 0\}. \tag{2.18}$$

This set of image pixels can be determined by evaluating (2.9), or more generally (2.5), for an assumed target trajectory.

In general, with image data modelled by (2.5), the target detection problem can be expressed as a binary hypothesis testing problem where the signal vectors $\mathbf{s}_1$ and $\mathbf{s}_0$ are complicated functions of the background clutter and target signals. However, if the target is additive to the background clutter and the sensor can be modelled as a linear system, then the sensor image data modelled by (2.8) can be expressed as

$$I[x_j, y_k, t_l] = \begin{cases} b_i[x_j, y_k, t_l] + t_i[x_j, y_k, t_l] + n_i[j, k, l] & \text{in the presence of a target, or} \\ b_i[x_j, y_k, t_l] + n_i[j, k, l] & \text{in the absence of a target} \end{cases} \tag{2.19}$$

where

$$\begin{aligned} t_i &= \textstyle\iiint A(t)\, p(x - x_t(t), y - y_t(t)) dx\, dy\, dt \\ b_i &= \textstyle\iiint I_b(x, y, t) * p(x, y) - \alpha I_b(x_t(t), y_t(t)) p(x - x_t(t), y - y_t(t))\, dx\, dy\, dt \text{ and} \\ n_i &= N[j, k, l] \end{aligned} \tag{2.20}$$

The target detection problem can then be expressed in terms of the binary hypothesis testing problem in (2.14) with

$$\begin{aligned} \mathbf{s_1} &= \mathbf{t} + \mathbf{b} \text{ and} \\ \mathbf{s_0} &= \mathbf{b} \end{aligned} \tag{2.21}$$

where $\mathbf{t} \equiv \{t_i \mid i = 1, 2, \ldots n\}$, $\mathbf{b} \equiv \{b_i \mid i = 1, 2, \ldots n\}$, and $\mathbf{n} \equiv \{n_i \mid i = 1, 2, \ldots n\}$ correspond with an arbitrary ordering of the observation set $\mathbf{y}$.

The two hypotheses are

- H_1: a target with amplitude $A(t)$ and trajectory $(x_t(t), y_t(t))$ is present, and
- H_0: a target with amplitude $A(t)$ and trajectory $(x_t(t), y_t(t))$ is not present.

This problem can then be evaluated for a number of hypothesized target trajectories. This is a multiple hypothesis approach to the composite hypothesis testing problem of detecting a known multidimensional signal with unknown, continuous, real-valued parameters.

There are a variety of noise sources associated with the sensor photodetection process [62], but the detector noise $\mathbf{n}$ in (2.14) is usually modelled as a sum of the photon shot noise associated with the random photon arrival time, and the thermal noise in the detector electronics [25]. In general, the detector noise level is very low and is not the limiting factor in detection performance. Consequently, the detector noise process is approximated by a Gaussian iid noise process which defines the *noise floor* for the detection problem [25, 33]. A notable exception to this practice is the model for detector noise in a photon limited application such as the wide field-of-view telescope discussed in Chapter 1. In this case the photon noise limits the detection performance and the detector noise is modelled as the sum of a Poisson random shot noise process and a Gaussian random noise process [33, 62].

It is well known that the optimal coherent detector ($\mathbf{s}_1$ and $\mathbf{s}_0$ both completely known and deterministic) for (2.14) with iid Gaussian noise is the correlation detector or matched filter [63, 64]

$$\delta(\mathbf{y}) = \begin{cases} 1, & \sum_{i=1}^{n} s_i^{\star} y_i > \tau \\ r, & \sum_{i=1}^{n} s_i^{\star} y_i = \tau \\ 0, & \sum_{i=1}^{n} s_i^{\star} y_i < \tau \end{cases} \tag{2.22}$$

where $s_i^{\star} = s_{1i} - s_{0i}$. This detector structure can be easily implemented by n'th sample of the output of a linear digital filter to a fixed threshold τ. The detection performance of this detector is determined by the parameter

$$\theta = \frac{(\mathbf{s}_1 - \mathbf{s}_0)^2}{\sigma^2} \tag{2.23}$$

where $\frac{(\mathbf{s}_1 - \mathbf{s}_0)^2}{\sigma^2}$ is the ratio of average signal power to average noise power in the output of the matched filter detector, the output signal-to-noise ratio (SNR).

The matched filter has the property that its output signal has the highest SNR of any linear filter with the same observation set. This result applies to the non-Gaussian iid noise case as well, if the noise has zero mean and a finite covariance [63]. The SNR can be viewed as a measure of the *distance* between the signals $\mathbf{s}_1$ and $\mathbf{s}_0$.

$$d^2 = ||\mathbf{s}_1 - \mathbf{s}_0||^2 \tag{2.24}$$

Thus, an increase in the SNR can be viewed as an increase in the distance between the two signals, suggesting an improvement in signal detection performance. It is intuitively reasonable that improving the signal-to-noise ratio will improve detection performance for the non-Gaussian iid noise case as well [63].

The detection optimality of the matched filter structure in terms of a Bayes, minimax or Neyman-Pearson criterion is dependent on the Gaussian noise assumption but the matched filter structure is a reasonable engineering approximation for the non-Gaussian case as well. The optimal detector structure for the non-Gaussian case is generally nonlinear but it can sometimes be modelled as a point non-linearity followed by a matched filter structure. In general, the use of the matched filter structure for SNR optimization is a common and generally accepted engineering technique even if the noise is non-Gaussian [64]. This structure is practical to implement and simplifies the corresponding performance analyses. This linear approximation is inappropriate, however, when the noise process contains impulsive non-Gaussian components.

In general, the background clutter signal $\mathbf{s}_0$ is unknown and is more accurately modelled as a random noise process. Consequently, the largest gains in target detection performance are achieved by estimating and suppressing the background clutter prior to signal detection. The optimal *linear* filter for SNR optimization in preprocessing the image data for detection is the linear matched filter. It is well known that simple, easily analyzable detector structures can be designed for the detection of a known signal in independent, identically distributed or Gaussian noise. Thus, it is desirable that the residual background clutter after estimation be uncorrelated and/or Gaussian.

2.2.2 Background Clutter Suppression

With the signal quality of modern electro-optical sensors, the effects of optical background clutter are a much greater impediment to target detection than the effects of sensor noise (with the Wide-Field-of-View Telescope Application of Chapter 1 being a notable exception). Consequently, the greatest single-frame detectability gains are achieved by attenuating the background clutter signal. The following algorithms are applicable to the class of image sequences which can be modelled by targets which are additive to the background signal.

Background clutter suppression algorithms attempt to estimate the background clutter image signal $b[\mathbf{x}, t]$ and subtract it from the received image sequence to yield a preprocessed image sequence with observations

$$\begin{aligned} \mathbf{y} &= \mathbf{t} + \mathbf{n}^\star \quad \text{under } H_1 \\ \mathbf{y} &= \mathbf{n}^\star \qquad\quad \text{under } H_0 \end{aligned} \tag{2.25}$$

where

$$\mathbf{n}^\star = (\mathbf{b} - \hat{\mathbf{b}}) + \mathbf{n} \tag{2.26}$$

If the background clutter estimate $\hat{b}[\mathbf{x}, t]$ is successful in capturing the spatio-temporal behaviour of $b[\mathbf{x}, t]$, then the residual noise term $\mathbf{n}^\star$ is approximately iid, and the target detection problem is reduced to the standard problem of detecting a deterministic signal in additive iid noise [63].

The ultimate goal of any preprocessing function is to improve the achievable target detection performance. Given complete *a priori* knowledge of the joint probability distribution of the image observations under each hypothesis in (2.21), an optimal, usually non-linear, transformation of the image observations could be derived. However, in general, the background clutter signal is not known *a priori* and linear preprocessing filters are designed to estimate and decorrelate (whiten) the background clutter signal. An ideal clutter suppression algorithm would be capable of transforming an arbitrary inhomogeneous, non-stationary background into an approximately stationary, homogeneous, iid Gaussian noise innovations background without reducing the effective target signal power. In practice, preserving the target signal energy is the critical consideration for weak targets.

Temporal filter algorithms

Temporal filter algorithms exploit the strong temporal correlation between observations of successive image frames [42, 41, 23, 66, 67]. However, the performance of these algorithms is dependent on either a stationary image background or precise image registration [68]. The simplest example of this class of algorithms is the simple frame difference. Image backgrounds which are nearly stationary (the background motion over the frame integration period is small relative to the total field-of-view) with slowly-varying statistics can be effectively suppressed by a sequential combination of adaptive discrete frame registration and linear filtering. In general however, the image background is dynamic and statistically non-stationary.

Spatial filter algorithms

Spatial filter algorithms estimate the background clutter signal directly from the image observations [69, 70, 23, 66, 67] or from a target free reference image [71, 72]. Previous frames in the image sequence or images from different spectral bands of a multispectral sensor are often suitable reference images. Many of these algorithms are adaptive and assume no knowledge of the background statistics. Adaptation to local characteristics within the scene is achieved by locally estimating the background image covariance matrix and then inverting the matrix to compute linear filter weights. This approach is optimal in the sense that no other linear filter can achieve higher clutter suppression, but is computationally very intensive.

Three-dimensional matched filter algorithms

Reed et al. derived the optimal three-dimensional, linear, matched filter for point-source targets of known velocity in image sequences with additive background clutter and noise [25, 30]. This filter is optimal in the sense that no other linear filter can achieve a greater signal-to-noise ratio in the filter output signal. Although the derivation of this filter assumes complete knowledge of the target, background and noise signals, it specifies the maximum achievable signal-to-noise ratio and demonstrates how the characterisitics of the target, background clutter, sensor optics and noise affect filter design and performance. The three-dimensional matched filter can be viewed as the optimal linear preprocessor for target detection when the target trajectory, target, background and noise signals are completely known.

Reed et al. implemented their three-dimensional matched filter in the Fourier transform domain. Their approach is to transform the image sequence, apply a matched filter tuned to the frequency domain signature of the moving target, inverse transform the filtered data and then threshold the time domain output for detection. The filter uses the assumed target velocity to coherently integrate the target energy along a linear, constant velocity trajectory. Thus, the filter can be viewed as a sliding-window correlation of the target profile as it moves through the image sequence volume, effectively integrating the target energy across space and time. The inverse transform of the result has an enhanced signal-to-noise ratio with peaks representing the initial position of targets with the assumed target velocity.

The primary advantages of filtering in the Fourier domain are the ability to simultaneously detect all targets with the same Fourier signature, moving at the same velocity in the imaged scene and to accomodate non-integer target velocities which are problematic in equivalent time-domain techniques. The primary drawback of this approach is the requirement that the filter be matched to a specific velocity profile of a known target moving at a known speed in a particular direction. The effects of velocity mismatch for these filters are dramatic, particularly for higher target velocities.

This problem can be partially overcome by designing a bank of filters to cover the desired range in speed and direction. If the target intensity is reasonably constant over time, then the SNR performance of a perfectly matched three-dimensional filter increases linearly with the number of frames processed. If however, the target intensity varies, then the three-dimensional filter is simply equivalent to a coherent addition of the spatial matched filter responses of each frame in the sequence.

Reed et al. applied the results of their three-dimensional matched filters to the development of a recursive moving target indicator (RMTI) algorithm with greatly reduced computational complexity [39]. The RMTI algorithm exploits the fact that the difference between the two-dimensional Fourier transforms of successive images is a periodic function. By shifting the two-dimensional Fourier transform of each image in phase, using the known target velocity, the target energy can be coherently summed as a weighted sum of two-dimensional Fourier images. A two-dimensional matched filter for the target signature can then be applied and the result inverse transformed as before.

The advantage of this approach over the full three-dimensional matched filter is the reduction in computation and data storage afforded by the reduction in the dimensionality of the required Fourier transforms. The RMTI algorithm can process each frame recursively without the cost of computing a three-dimensional Fourier transform for each new frame and the cost of the memory required to store the preceding N frames. However, the RMTI approach requires a prewhitened image sequence [71, 72] in order to fully achieve its predicted performance. As with the three-dimensional matched filter, this approach is critically dependent on precise knowledge of the target velocity relative to the image background and is subject to a dramatic loss of performance with velocity mismatch.

Summary

The development of high performance clutter suppression and target enhancement algorithms with affordable levels of complexity is critical to improving the performance of surveillance systems for detecting weak targets. These algorithms are often refered to as prewhitening algorithms because of the critical need to decorrelate the image background observations prior to detection. The assumption of iid image observations is critical to the successful application of standard signal detection techniques. Recently, three-dimensional spatio-temporal and four-dimensional spatio-temporal/multispectral filters have been proposed and analyzed [73–77]. These algorithms incorporate models of the target signal and the image background and *a priori* knowledge of the target and background statistics to maximize the signal-to-noise ratio of the filtered image data.

2.2.3 Detect-Before-Track Algorithms

'Detect-before-track' algorithms can be characterized by decision rules which make *hard* detection decisions after every frame. Typically the incoming image sequence is preprocessed with a filter designed to suppress the background clutter signal and then the optimal Bayes, minimax or Neyman-Pearson decision rule is applied to the pixels in the preprocessed image. The detection model for these algorithms is given by (2.25) with the image observation set $\mathbf{y}$ for image frame l given by

$$\mathbf{y} = \{I[x_j, y_k, t_l] \mid w_{j,k,l}(x_t(t), y_t(t)) \neq 0 \, (l-1)\Delta t \leq t \leq l\Delta t\} \qquad (2.27)$$

In practice, the target trajectory is unknown and it is computationally infeasible to design an optimal detector for the continuum of possible target positions. Consequently, a suboptimal set of discrete target positions is tested with a concomitant loss in detection performance. If the target is assumed to be located at the centre of one of the detector pixels, a single decision rule can be designed to test for the presence of a target centred on each pixel in the image frame. The performance loss incurred by assuming the target is centred on a detector pixel can be evaluated as a function of the sensor point-spread function and the detector response. This loss can be reduced by subsampling the pixel surface area and designing a bank of detectors for a finite number of subpixel target locations.

The simplest example of this class of algorithms is the application of a single fixed threshold to each pixel in the image frame. This is the optimal detection procedure for the detection of a constant intensity point-source target in an additive iid Gaussian noise background under ideal imaging conditions

$$\begin{aligned} Y &\sim N(\mu_1, \sigma^2) \text{ under } H_1 \\ Y &\sim N(\mu_0, \sigma^2) \text{ under } H_0 \end{aligned} \tag{2.28}$$

where $N(\mu, \sigma^2)$ is the standard Gaussian probability density function with mean μ and variance σ^2. As discussed in section 2.1, under ideal imaging conditions, the target image consists of a single pixel and is independent of the subpixel location of the target on the sensor focal plane. This implies that a single optimal test can be designed to independently detect the presence of a target in each pixel in the image frame.

Since the image observations for this problem are realizations of iid Gaussian random variables, the optimal decision rule can be expressed as a log likelihood decision rule (2.16) with

$$L(y) = \frac{N(\mu_1, \sigma^2)}{N(\mu_0, \sigma^2)} \tag{2.29}$$

Those pixels which exceed the decision threshold are declared target detection observations. Although this approach is computationally very simple, acceptable performance in terms of the detection error probabilities α and β for practical imaging systems is dependent upon a high intrinsic single pixel signal-to-noise ratio or approximately ideal imaging conditions.

Change Detection Algorithms

When the signal under the target present hypothesis H_1 is completely unknown but the signal under the target absent hypothesis H_0 can be estimated, decision rules can be derived for the binary hypothesis problem [25, 78, 79]

- $H_1 : \ f(\mathbf{y}) \neq f_{H_0}(\mathbf{y})$
- $H_0 : \ f(\mathbf{y}) = f_{H_0}(\mathbf{y})$.

This model assumes that any image signal which is statistically different than the estimated image signal is a target signal. Consequently, the performance of these techniques is highly dependent on an adequate estimate of the image background. The probability distribution of image observations in the absence of a target is ususally estimated from reference images such as previous frames in the image sequence or correlated images in other spectral ranges. These techniques can also be used as a coarse first stage detection filter in a multi-stage detection algorithm (see section 3.3.1).

Optimal decision rules for the composite two-sided detection problem defined above do not exist in general [64]. However, if the target signal model is known and it is known *a priori* that the target is not initially present, change detection algorithms can be designed which exploit target free reference images to detect the initial appearance of the target. For example, Reed et al. have developed a constant false alarm rate (CFAR) detection algorithm which exploits target free reference images to detect the initial appearance of a known target [71, 72, 80]. They claim that their algorithm can achieve as much as a 10 dB improvement in the effective detection signal-to-noise ratio with a single correlated reference image. In general, the performance achieved by this approach improves with an increase in the number of reference images and is dependent on the correlation between the reference images and the current image.

Track Assembly

For some applications, frame by frame detection is insufficient to meet system detection performance requirements. For these cases, track assembly algorithms have been developed which project preprocessed images onto a single target track image for detection. By exploiting target trajectory properties such as motion continuity, smoothness or temporal persistence, heuristics can be proposed to suppress spurious detections [23, 24, 29, 54, 42]. These algorithms are often referred to as *streak* detectors since they typically detect single pixel detections which form linear or curvilinear trajectories in the target track image. Although these algorithms are computationally attractive their ability to detect weak targets is suboptimal.

The optimal projection scheme for multidimensional Neyman-Pearson detection of known targets in iid noise (see (2.25)) was derived by Chu [29] providing a baseline for comparing the performance of the heuristic algorithms described above. Chu analyzed the performance loss ($\simeq 3\,$dB for an unprojected data SNR of 10dB) of the optimal projection scheme relative to a full multidimensional matched filter (see section 2.2.5). A maximum value projection scheme was also proposed which is a good approximation to the optimal projection scheme for unprojected data with a SNR greater than 6 dB.

2.2.4 Frequency Domain Algorithms

Frequency domain techniques exploit differences in the spatiotemporal frequency spectra of the target and background clutter/noise to detect moving targets. In general, moving point-source targets tend to have a higher spatial frequency content, due to their limited spatial extent, and a higher temporal frequency content, due to their relative motion, than do background clutter and noise signal sources. The following algorithms derive optimal decision rules for target detection in the Fourier domain.

Warren has proposed a sequential generalized likelihood ratio detector, conditioned on target amplitude and location, for small targets in homogeneous stationary additive background clutter [81]. This detector is implemented in the Fourier domain by sequentially processing the spatial Fourier transform of the incoming image frames. Each spatial frequency component is effectively processed independently in time. SNR enhancement is achieved by separating the target and clutter signals in the spatial frequency domain and by estimating the clutter mean and covariance for clutter suppression.

Porat and Friedlander proposed a class of frequency domain directional filters to enhance targets with an assumed linear, constant velocity trajectory [37]. A directional filter is designed for each candidate trajectory and the resulting filter bank used to test an image sequence for the presence of targets with the assumed characteristics. The output of each filter is an image which corresponds with a two-dimensional slice of the spatio-temporal image volume.

The output of those filters for which a target is detected are processed in the time-domain to estimate the trajectory of each target with a given directionality. This approach has the advantage of being independent of target position. Each directional filter will detect the presence of any target with the assumed target velocity. The primary difference between this approach and the three-dimensional matched filter of section 2.2.2 is that the target detection decisions are made in the Fourier domain. A similar approach for constant intensity targets was proposed by Bruton and Bartley [27].

2.2.5 Track-Before-Detect Algorithms

For low contrast targets (preprocessed SNR ¡ 10 dB), single frame and track assembly algorithms are often insufficient to meet detection performance specifications. The 'track-before-detect' approach to the MFTDT problem is to estimate or track a large number of possible candidate target trajectories without initially declaring the presence of a target. The likelihood that a target with the assumed characteristics is present is then evaluated for each candidate trajectory.

Given an assumed target trajectory, a maximally overlapping set of target trajectory pixels can be found, and a decision rule may be designed for optimal detection conditioned on the assumed target trajectory. The following 'track-before-detect' algorithms employ detection decision rules for the image observation set $\mathbf{y}$

$$\mathbf{y} = \{I[x_j, y_k, t_l] \mid w_{j,k,l}(x_t(t), y_t(t)) \neq 0\}. \tag{2.30}$$

for an assumed target trajectory $(x_t(t), y_t(t))$. The increase in detection sensitivity over single frame detection algorithms is a direct function of the increased sample size of the image observation set.

Velocity Filters

The velocity filter approach, first proposed by Mohanty [22], is to exhaustively test a complete set of time-domain candidate target trajectories for the presence/absence of a target. Given statistical models for image observations in the presence and absence of a target, the optimal detector for each trajectory can be designed using classical detection theory. For the case of a known target in an additive Gaussian noise background, the velocity filter approach is equivalent to a time-domain version of the three-dimensional matched filter described in section 2.2.2 [33, 32].

Given the image sensor's response to an arbitrary point-source target (2.9), an optimal set of image observations $\mathbf{y}$ can be selected for the binary hypothesis testing problem of (2.25). This amounts to binary hypothesis testing of the collection of image pixels $\mathbf{y}$, against a suitable threshold, τ, for the absence (H_0) or presence (H_1) of a target. For example, the decision rule for a constant intensity target in zero mean, additive, iid Gaussian noise can be expressed as a log-likelihood decision rule (2.16) with

$$L(y_i) = \frac{N(Aw_{j,k,l}, \sigma^2)}{N(0, \sigma^2)}. \tag{2.31}$$

It is assumed that the observations along a true target trajectory and the observations along hypothesized trajectories through the image background can be characterized as samples from one of two different probability distributions with known parameters.

$$\begin{aligned} H_0: \quad & \mathbf{Y} \sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \Theta_0) \text{ For false trajectories.} \\ H_1: \quad & \mathbf{Y} \sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \Theta_1) \text{ For true target trajectories.} \end{aligned} \tag{2.32}$$

Note that this is a batch processing technique. The decision rule is designed to make an optimal decision for the image observations received over K frames. The same number of observations are used in evaluating each candidate trajectory. Hence, these hypothesis tests are referred to as fixed sample size (FSS) tests.

Typically, targets are assumed to follow linear, constant velocity trajectories, at least over a short time interval. This assumption is reasonable for many targets, if the sensor frame rate is sufficiently high, and effectively constrains the trajectory search space to be a function of three target parameters

1. initial position,

2. speed, and

3. angular direction in the focal plane.

These three parameters define a subspace of $\Re^3$ where a *real* target trajectory is defined by a single point (p, s, d). In principle, there are an infinite number of *real trajectories* which satisfy any *a priori* constraints on the target's speed, direction or initial position. The velocity filter approach is thus a multiple hypothesis approach to the composite hypothesis testing problem of detecting a known target in additive noise with unknown parameters. Each binary hypothesis test is of the form (2.25) with an assumed target velocity and initial position.

Intrinsically, testing a continuous range of assumed target positions and velocities requires an infinite number of candidate trajectories. In practice, the candidate trajectory search space can be dramatically reduced by an intelligent partitioning of the velocity search space [31, 43]. Given a model such as (2.5) or (2.7) for the image formation process, the image plane response for a target with an arbitrary real trajectory (p, s, d) can be evaluated. Then, by defining a maximum acceptable signal-to-noise ratio loss factor, suboptimal partitions of the trajectory search space $(p, s, d) \in \Re^3$ can be designed [31, 43]. The resulting candidate trajectory set (velocity filter bank) minimizes the number of candidate trajectories while limiting the performance loss due to parameter mismatch.

There have been other attempts to reduce the effective trajectory search space while maintaining desired performance specifications. In the following, two algorithms will be described which approximate the behaviour of the exhaustive approach and reduce the computational complexity without unduly sacrificing detection performance. The first algorithm is a tree-structured multistage hypothesis test algorithm and the second is an application of the dynamic programming algorithm. Both algorithms assume the existence of reasonable statistical models for the image trajectories in the presence or absence of a target and provide explicit mechanisms to compute the key performance specifications: the detection error probabilities of false alarm and missed detection.

Tree-structured Multistage Hypothesis Testing

Recently Blostein and Huang proposed an alternative implementation of the velocity filter approach which offers a substantial reduction in computational complexity [40]. Their approach exploits two key observations. The first is that the classical fixed sample size hypothesis test for each candidate trajectory can be computationally inefficient. The second is that a large number of candidate target trajectories pass through the same pixels in the first few image frames and hence, under ideal imaging conditions, share common image observations. Their MSHT object detection algorithm exploits a sequential decision rule, a truncated SPRT, to sequentially prune a dense tree of linear, constant velocity candidate trajectories.

The standard velocity filter implements a fixed sample size (FSS) hypothesis test which has a fixed computational cost. An alternative approach is to apply a sequential decision rule to the evaluation of candidate trajectories. With a sequential decision rule, the sample size of the hypothesis test is a random variable. Thus, the computational cost is also variable.

The classical sequential decision rule is Wald's sequential probability ratio test (SPRT) [82]. An SPRT compares the partial sums of the likelihood or log-likelihood ratio statistic to a pair of thresholds after each sample (observation). If the test statistic exceeds the upper threshold, hypothesis H_1 is accepted, if the test statistic falls below the lower threshold, hypothesis H_0 is accepted, but if the test statistic falls between the two thresholds, then the detection decision is deferred for another sample. It can be shown that a pair of thresholds can be designed to meet any specified detection error probabilities (α and β) and that the resulting test will terminate.

The advantage of the sequential probability ratio test is that it is data adaptive. In contrast with the classical FSS test, the number of observations n is a function of the actual distribution of the received observations. The SPRT takes as many samples as required to make a decision which satisfies the specified decision error probabilities. This property is particularly advantageous when the decision error probabilities are unequal.

For example, typical detection performance specifications for a single candidate trajectory require a much lower probability of false alarm than probability of missed detection. Thus, fewer samples are required to meet the performance specifications for a decision to accept the target absent hypothesis H_0 than are required to accept the target present hypothesis H_1. The sequential test fixes the desired performance specifications and allows the test sample size to vary as a function of the received observations while a typical FSS test (Neyman-Pearson) maximizes the probability of detection for a fixed number of samples and a given false alarm rate. In fact, it can be shown that under H_0 or H_1, a sequential probability ratio test has the minimum average sample size of any likelihood ratio test with detection error probabilities no greater than α and β [63].

The SPRT has two drawbacks which limit its practicality. The first is that although the test is guaranteed to terminate, the maximum sample size is unbounded. This can occasionally result in undesirably long tests. The second disadvantage is that the SPRT is sensitive to parameter mismatch. A modest parameter mismatch can cause the average test length of an SPRT to exceed the length of an equivalent FSS test.

As a practical compromise, Blostein and Huang proposed the use of Truncated Sequential Probability Ratio Tests (SPRT) [83]. The truncated SPRT is a sequential decision rule that is truncated after a finite number of stages. It is a robust approximation to Wald's SPRT and can be viewed as a mixture between a FSS test and an SPRT. The performance of a truncated SPRT is a compromise between the minimum average sample size of an SPRT and the test robustness to parameter mismatch of an FSS.

Blostein and Huang describe the truncated SPRT as an important example of a multistage hypothesis test.

Definition: a **multistage hypothesis test (MSHT)** is any sequential test with a finite number of stages, K.

$$\sum_{j=1}^{i} x_j \begin{cases} \geq a_i & \Rightarrow \text{choose } H_1 \\ \leq b_i & \Rightarrow \text{choose } H_0 \\ \in (\mathbf{b}, \mathbf{a}) & \Rightarrow \text{take another sample} \end{cases}$$

for stage i $1 \leq i \leq K-1$, and

$$\sum_{j=1}^{K} x_j \begin{cases} \geq a_K & \Rightarrow \text{choose } H_1 \\ < \text{otherwise} & \Rightarrow \text{choose } H_0 \end{cases} \tag{2.33}$$

where a_i and b_i are the decision thresholds for test stage i. Closed-form expressions for the performance of a multistage hypothesis test with iid observations were derived by Blostein and Huang. These expresssions were used to predict the performance of a multistage hypothesis test with iid Gaussian observations and to estimate the processing and memory requirements of their MSHT object detection algorithm.

Blostein and Huang exploit the smaller average sample size of the sequential decision rule to reduce the computational demands of the velocity filter approach. The sequential test rapidly identifies candidate trajectories where the target is clearly present or absent, allowing the system to focus its resources on evaluating trajectories where the observations are more ambiguous. The computational savings of this approach approach are most dramatic when the candidate trajectory set can be structured as a dense search tree.

Under ideal imaging conditions, the focal plane image of a point-source target is also a point. Thus, the detected target energy is focused on a single detector pixel and the intensity of the resulting image pixel is independent of the sub-pixel location of the target. Thus, the target signal for an arbitrary target trajectory is completely determined by the sequence of detector pixels upon which the target is focused in subsequent frames. This reduces the continuum of target positions and velocities to a discrete set of candidate trajectories quantized by the detector pixel array.

For target velocities of approximately 1 pixel/frame or less, the discrete candidate trajectory set can be structured as a dense tree of linear, constant velocity trajectories, as illustrated in figure 2.1. Those trajectories that pass through the same pixels in the first few image frames share the same initial observations. Thus, if the same decision rule is applied to all the candidate trajectories, an early decision to reject the target present hypothesis can simultaneously reject all candidate trajectories with common observations up to and including the decision stage. Thus, a sequential decision procedure can achieve a substantial reduction in computational complexity, without an associated performance loss, by sequentially pruning the dense search tree.

Figure 2.1: Tree-Structured Trajectory Search

Dynamic Programming Algorithms

The dynamic programming algorithm (DPA) is a recursive solution to the problem of finding an optimal sequence of system states in the temporal evolution of a dynamic system. The DPA was originally developed to solve problems in multistage optimal control [84] but has found widespread application in optimal estimation and signal processing. Recently, there has been significant interest in applying the DPA to the detection and tracking of low observable, moving point-source targets in a sequence of digital images [26, 28, 34, 36, 85].

The dynamic programming approach to the MFTDT problem is to find the candidate target trajectory T_{opt} which maximizes the *a posteriori* probability, conditioned on all the received image data, that the candidate trajectory is a true target trajectory. This problem can be expressed more succinctly as the search for the sequence of target states $T_k \equiv [ts_1, ts_2, \ldots, ts_k]$ which maximizes the likelihood ratio

$$\frac{\Pr(H_1 \mid Y_k)}{\Pr(H_0 \mid Y_k)} \tag{2.34}$$

where $\Pr(H_i \mid Y_k)$ is the probability that the image observations $Y_k \equiv [y_1, y_2, \ldots, y_k]$ along the candidate trajectory T_k are distributed according to the signal model of hypothesis H_i (recall (2.14)), conditioned on the image data received up to and including stage k, Y_k.

In order to place the target detection and tracking problem in a dynamic programming framework, the image sequence is divided into stages of length G frames/stage. The set of target states is defined as the set of possible target paths through a given stage from every pixel in image frame i to every pixel in image frame $i + G - 1$. If the target trajectories are required to be linear, then the number of states/stage is equivalent to the number of hypothesis tests required for a FSS velocity filter over G image frames.

The system model for the dynamic programming framework can be summarized by a state transition equation and a model for the image observations under the target present/absent hypotheses, given by (2.14). The state transition equation is usually a first-order Markov model relating each target state at stage $k + 1$ to a prior target state at stage k through a state transition function $st()$

$$ts_{k+1} = st(ts_k, n_k) \tag{2.35}$$

with an additive iid noise process n. If the probability of occurrence of any state ts_{k+1} is dependent only on the previous state ts_k, then the transition from a state at stage $k+1$ to a state at stage k does not affect the optimality of previous state transitions and the maximization of (2.34) can be divided into two parts such that

$$\max_{T_k} \frac{\Pr(H_1 \mid Y_k)}{\Pr(H_0 \mid Y_k)} = \max_{ts_k} \max_{T_{k-1}} \frac{\Pr(H_1 \mid Y_k)}{\Pr(H_0 \mid Y_k)}. \tag{2.36}$$

The inner maximization is a maximization of the *a posteriori* probability density function for the candidate trajectory up to stage $k-1$, conditioned on the observations received by image frame k. This function can be used to define a *merit function* $S(T_k)$ to evaluate the candidate trajectories (states) at the end of each stage.

Referring to (2.36), and using Bayes formula

$$\frac{\Pr(H_1 \mid Y_k)}{\Pr(H_0 \mid Y_k)} = \frac{\Pr(y_k \mid ts_k)}{\Pr(y_k \mid H_0)} \frac{\Pr(ts_k \mid ts_{k-1}) \Pr(H_1 | Y_{k-1})}{\Pr(H_0 | Y_{k-1})}. \tag{2.37}$$

leads to a recursive formulation for the merit function of the form

$$S(T_k) = \ln[\frac{\Pr(y_k \mid ts_k)}{\Pr(y_k \mid H_0)}] + \ln[\Pr(ts_k \mid ts_{k-1})] + S(T_{k-1}). \tag{2.38}$$

Note that the merit function for each state can be propagated from stage to stage. The optimal trajectory at a given stage can be found by searching the set of target states for the state with the maximum merit function. If a record of the optimal state transitions at each stage is maintained, then the optimal trajectory can be found by tracing the state transitions backwards, from the optimal state at any given end stage.

In fact, one need not choose the single trajectory which maximizes the likelihood ratio of (2.34). Instead all trajectories which exceed a threshold designed to meet specified detection error probabilities can be declared target trajectories. However, this process introduces additional processing problems.

The dynamic programming algorithm has a problem with strong cross-correlations between adjacent target states. A single strong target trajectory will generate strong responses for similar trajectories, reducing the algorithm's sensitivity to neighbouring weak targets. Thus in order to relax the strict maximization, a designer has to provide heuristic methods for evaluating and maintaining the other target trajectories which are generated.

In principle, the DPA with a single stage is equivalent to the velocity filter proposed by Mohanty [22]. The merit function in this case reduces to an evaluation of the likelihood ratio of the observations received along the candidate trajectories used in the hypothesis tests of the velocity filter. This comparison provides a vantage from which to analyze the computational reduction the DPA achieves by dividing the image sequence into stages.

Barniv derived an estimate of the number of operations required to implement the DPA on an M frame sequence of $N \times N$ images divided into M/G stages [85].

$$\text{Total number of operations} = 50\frac{M}{G}N^2(G+1)^2[2Gr_o + 13] \tag{2.39}$$

where r_o is the number of operations needed to evaluate the likelihood ratio $\frac{\Pr(y_k|H_1)}{\Pr(y_k|H_0)}$. From this expression one can derive the approximate relationship that a DPA with M/G stages requires $(G/M)^2$ times the number of operations required by an M frame velocity filter.

However, this reduction in computational complexity comes at a cost of reduced performance. The detection performance of the algorithm is manifested in the evaluation of the likelihood ratio for the image observations along the target states of each stage. Thus, the detection sensitivity of the DPA is directly dependent on G, the stage length.

The stage length is a key parameter in the performance vs computation tradeoff for the DPA. There is an optimal stage length G which minimizes the required computations for a desired level of detection performance. However, as the intrinsic signal-to-noise ratio decreases the optimal number of stages approaches 1. Researchers have found that the DPA is most effective when the intrinsic SNR is from $2 - 5dB$. The computational gain for weaker signals is negligible and there are computationally simpler algorithms which can attain the same detection performance for stronger signals.

2.2.6 Summary

The appropriate target detection algorithm for a given application is dependent on the required level of performance, as measured by the detection error probabilities α and β, and the acceptable computational load. The 'detect-before-track' algorithms are computationally simple but exhibit reduced senstivity to weak targets. Conversely, the 'track-before-detect' algorithms are computationally expensive but exhibit substantially improved detection sensitivity. Although the exhaustive velocity filter approach is optimal for the detection of known targets in additive Gaussian noise it is computationally unwieldy and the design of efficient velocity filter banks is a difficult problem. In practice, suboptimal 'track-before-detect' algorithms trade a loss in optimal detection performance for a substantial decrease in the computational load.

2.3 Multitarget Tracking and Data Association

Multitarget tracking (MTT) is an essential function of surveillance systems designed to identify individual targets in a noisy, cluttered, multitarget environment. Traditionally, target tracking in noisy, cluttered environments has been treated as a problem in associating detected target observations with target tracks in a dense multitarget environment. This problem is posed as a state estimation problem where the system state is augmented to account for uncertainty in the origin of the state observations. Any individual observation could have a false target detection or one of an unknown number of true targets, as its source. Thus, the performance of these techniques is fundamentally dependent on target detection performance. Recently, increasingly demanding MTT system requirements have fostered an interest in combining the target detection and tracking problems.

The function of a multitarget tracking algorithm is to estimate the *state* of each target in the sensor's field-of-view from the detected target observations produced by the sensor's signal processing unit (see figure 1.2). This task has to be accomplished in the presence of uncertainty in the origin and accuracy of the target observations.

In what follows, the MTT problem will be addressed in three parts: state estimation and prediction, data association (track formation and maintenance), and track confirmation and deletion.

2.3.1 Target State Estimation and Prediction

Targets are typically modelled as dynamic systems with a state-space representation. In this framework, the target *state* is a vector of parameters which characterizes an underlying model of the target dynamics [7]. The target state typically consists of kinematic parameters such as target position, velocity and acceleration; and auxiliary parameters such as signal strength and target spectral response. Multiple models of target dynamics may be required to account for different modes of target behaviour or target manoeuvres [5–7].

A typical target observation includes noisy measurements of the target's position and signal strength. Given a sequence of observations of an individual target, the target state can be estimated and predicted using standard state estimators such as a Kalman or fixed coefficient filter [6, 7]. These filters use estimates of the accuracy of the target observations to estimate the resulting accuracy in the state estimates. Modifications to these standard filters are often required to account for the possibility of misassociation [7].

The accuracy of the state estimates is dependent on

- the accuracy of the assumed model for target dynamics and observations,
- the accuracy and information content of the target observations, and
- the *purity* of the observation set used for state estimation.

where track *purity* is a measure of the number of observations which have been incorrectly associated with a given target track. The standard state estimation filters are designed to manage the uncertainty associated with the accuracy of the target observations. However, their performance degrades substantially in the presence of association errors. Association errors can corrupt the covariance estimate of the filter and lead to a loss of track or track deletion. In a dense multitarget environment, the success of the MTT algorithm in partitioning the set of target observations into subsets of observations, or target *tracks*, associated with individual targets is a critical factor in tracking performance.

2.3.2 Data Association

The formation of target tracks in a noisy, cluttered, multitarget environment is confounded by a significant degree of uncertainty in the origin of the target observations. Any individual observation could be from a false alarm, or from any of an unknown number of true targets and for any set of target observations, there is usually more than one plausible way to partition the observations into target tracks. This partitioning of the observation set or *data association* is the first task of any MTT system.

One can identify two distinctive data association tasks:

1. observation-to-observation association or track formation (initiation), and

2. observation-to-track association or track maintenance (update).

New observations are considered as updates to existing target tracks and as a source for the initiation of new target tracks. Typically, the data association process begins with a *gating* procedure designed to exclude consideration of improbable observation-to-track associations.

The gating procedure defines a region or *gate* around the predicted position of each target track. Only those observations which fall inside the gate are considered as candidates for association. Typically, the size of the gate is a function of the expected target dynamics, including manoeuvres, and the covariance estimate of the state estimation filter.

If a target observation falls inside the gate of a single target track, and is the only observation to do so, then the association task is complete. However, in a noisy, cluttered, multitarget environment, multiple observations may fall within the gates of each target track and each observation typically falls within the gate of more than one target track. Under these conditions, the gating procedure only partially resolves the data association problem.

There are three basic approaches to resolving the remaining ambiguity in observation-to-track association

- deterministic or non-Bayesian probabilistic algorithms (Type I),
- single-frame Bayesian probabilistic algorithms (Type II), and
- multiframe Bayesian probabilistic algorithms (Type III).

Type I algorithms view the data association problem as a multidimensional assignment problem [86]. They attempt to find a unique pairing of observations to tracks such that each track is updated by a single observation which minimizes a local or global cost function. The optimal solution to these problems is NP complete [86] so suboptimal solutions are often used. These algorithms make *hard* association decisions at the end of every frame and lack explicit means for incorporating the probability of misassociation in the subsequent state estimates. Thus, these algorithms perform poorly in the presence of false alarms or in a dense target environment.

The nearest-neighbour (NN) approach [6, 7, 9] is typical of this type of technique. An NN algorithm finds a set of unique observation-to-track associations which minimizes a measure of the *distance* between the expected target position and the observation accepted to update the track. Each track consists of a sequence of observations, one from each image frame, which are assumed to originate from a single target.

Type II algorithms incorporate all the observations in the target track gate in the subsequent state estimate. Each observation is associated with the track in proportion to their probability of association. This produces a probabilistic weighted sum of the gated observations as a state update observation. The probabilistic data association (PDA) and joint probabilistic data association (JPDA) filters [10] are classical examples of this type of all-neighbour algorithm. Typically a Type II algorithm has 2 to 10 times the computational requirements of a Type I algorithm.

Type III algorithms attempt to consider all possible association decisions over a number of frames and make *soft* (i.e. not irreversible) association decisions. Multiple hypotheses are maintained with the knowledge that the most likely hypothesis at a given stage may be the continuation of a less likely hypothesis from a previous stage. The computational cost of maintaining multiple association hypotheses can be an order of magnitude greater than the cost of a Type II algorithm, but a Type III algorithm finds the *best* (maximum likelihood) data association conditioned on **all** the received observations. In practice, the optimal maximum likelihood approach must be sacrificed, as only a finite number of hypotheses can be maintained.

The multiple hypothesis tracking (MHT) approach [7, 8, 11] is a method of evaluating competing data association hypotheses by evaluating the *a posteriori* probabilities of their validity, conditioned on the received observations. Each hypothesis consists of a set of observation-to-track associations, where any individual observation-to-track association may belong to more than one hypothesis. New hypotheses are generated from old hypotheses after every observation update. Thus, each new hypothesis has a well-defined prior probability, the *a posteriori* probability of its parent hypothesis. This allows the likelihood of competing hypotheses to be evaluated recursively, conditioned on the most recent observation update. In principle this is an *optimal* Bayesian approach. However, in practice only a finite number of hypotheses can be maintained, so hypotheses with a small but nonzero probability are deleted and *similar* hypotheses are merged.

Defining the sequence of observation updates as $D_k \equiv \{d_1, d_2, \ldots, d_k\}$, where every observation update consists of a set of detected target observations d_k, the likelihood that the data association hypothesis H_l is valid, can be evaluated recursively by applying Baye's rule to

$$\Pr(H_l \mid D_k) = \frac{\Pr(d_k \mid H_l)\Pr(H_l \mid D_{k-1})\Pr(D_{k-1})}{\Pr(D_k)} \tag{2.40}$$

Letting

$$\overset{\prime}{\Pr}(H_l \mid D_k) = \Pr(d_k \mid H_l)\Pr(H_l \mid D_{k-1}) \tag{2.41}$$

(2.40) can be written as

$$\Pr(H_l \mid D_k) = \frac{\Pr'(H_l \mid D_k)}{\sum_l \Pr'(H_l \mid D_k)} \tag{2.42}$$

Thus, an evaluation of all the data association hypotheses relating the detected target observations d_k to the current set of target tracks is required to update the likelihood function for each hypothesis. In practice, many if not most of these hypotheses will be deleted. Thus, as the number of detected target observations increases this approach becomes extremely inefficient in its use of computational resources.

Blackman et al. have recently proposed an efficient, track-oriented implementation of multiple hypothesis tracking for dense, multitarget environments [35]. The Structured Branching (SB) implementation of multiple hypothesis tracking replaces the evaluation of competing data association hypotheses (2.40) with an evaluation of target track hypotheses T_i

$$\Pr(T_i \mid Y_k) = \frac{\Pr(y_k \mid T_k) \Pr(T_k \mid Y_{k-1}) \Pr(Y_{k-1})}{\Pr(Y_k)} \tag{2.43}$$

where $Y_k \equiv \{y_1, y_2, \ldots, y_k\}$ is the set of observations for a target following track T_i. In practice, there may not be a detected target observation corresponding with each y_i. Those target track observations y_i for which there is no corresponding detected target observation are considered as *missed detections*.

A sequential probability ratio test is applied to each target track as a binary hypothesis test, with hypotheses

- H_1: target present, $Y_i \sim \Pr(y_i \mid H_1)$ and
- H_0: target absent, $Y_i \sim \Pr(y_i \mid H_0)$ (all observations are false alarms).

The sequential probability ratio test evaluates the log-likelihood ratio

$$L_{T_i}(Y_k) = \ln \left[\frac{\Pr(T_i \mid Y_k)}{1 - \Pr(T_i \mid Y_k)} \right] \tag{2.44}$$

which can be evaluated recursively as

$$L_{T_i}(Y_k) = L_{T_i}(Y_{k-1}) + \ln \left[\frac{\Pr(y_k \mid H_1)}{\Pr(y_k \mid H_0)} \right] \tag{2.45}$$

where

$$\frac{\Pr(y_k \mid H_1)}{\Pr(y_k \mid H_0)} = \begin{cases} \frac{(1-\beta) e^{-\frac{d^2}{2}}}{\alpha (2\pi)^{\frac{M}{2}} \sqrt{|S|}} & \text{if } y_k \text{ is a detection} \\ \beta & \text{if } y_k \text{ is a missed detection} \end{cases} \tag{2.46}$$

and

$$\frac{e^{-\frac{d^2}{2}}}{\alpha (2\pi)^{\frac{M}{2}} \sqrt{\mid S \mid}} \tag{2.47}$$

is the M-dimensional Gaussian probability density function for the observation residual of the Kalman filter, with residual covariance matrix S, used for target state estimation [7].

Instead of directly generating global data association hypotheses, the SB implementation generates and evaluates single-track hypotheses. Those target tracks which are confirmed by the SPRT are then used to generate and evaluate global data association hypotheses. By evaluating individual tracks prior to the formation of global hypotheses, the computational complexity of the multiple hypothesis tracking approach is substantially reduced.

Note that this approach differs from the multistage hypothesis test proposed by Blostein and Huang. It is making a probabilistic decision based on a model of the target state space dynamics whereas the multistage hypothesis test algorithm proposed by Blostein and Huang evaluates the observations in terms of a received signal model. The SB-MHT algorithm operates on *detected target observations* and its likelihood ratio is a function of the detector error probabilities and the residual covariance matrix of the state estimator.

Summary

The following general comments can be made regarding the suitability of the previous data association techniques for various target environments. Type I algorithms are appropriate in sparse target environments where the false target density is low. Their performance can be improved by introducing a limited branching process similar to the MHT approach [15] and by modifying the state covariance estimate with an estimate of the probability of misassociation [7]. For moderate target densities, Type III algorithms offer the best performance due to their ability to correct association errors with the reception of additional observations. However, as the target density increases computational limitations preclude the use of Type III algorithms. Under these conditions Type II algorithms may be successfully applied. Their simple recursions eliminate the overhead associated with multiple association hypotheses while the incorporation of multiple observations provides a measure of robustness to association errors. In extremely dense target environments it is difficult to maintain the identity of individual targets, hence, group tracking methods are often applied [7].

2.3.3 Track Life Stages: Initiation, Confirmation, Deletion

The level of uncertainty associated with existing tracks, or track *quality*, is continuously monitored throughout the *life* of a target. In general, target tracks are formed (birth), confirmed and updated (life), and eventually deleted (death) as targets are acquired and lost. Low quality tracks are deleted to reduce the number of false tracks and high quality tracks are used to estimate current target states and to predict future target positions. These predictions are then used to aid in data association for the next set of target observations as the processing cycle repeats.

Incoming observations are first considered as updates for existing target tracks. Observations which are not associated with existing tracks are then considered for the initiation of new target tracks. There are many heuristic algorithms used for track initiation and typically, these new *tentative* tracks have to be *confirmed* before they are used for state estimation.

Simple track confirmation rules can be defined by requiring that M of the last N observation sets contain observations correlated with the newly initiated track or that the likelihood or *score* function associated with the track hypothesis exceeds a certain threshold. The streak detectors and 'track-before-detect' algorithms discussed in section 2.2 are also well suited to this task. They combine the functions of target detection, track initiation and track confirmation. Similar heuristics can be defined for track deletion. Tracks are deleted when the track likelihood falls below a certain threshold or when the last N observation sets did not contain observations suitable for track update.

2.3.4 Summary

Traditionally target detection and tracking have been treated as separate problems. Recent demands for improved MTT system performance in increasingly challenging signal environments have driven the development of both multiframe detection and multiframe association algorithms which attempt to make optimal use of all available information. These algorithms make detection and association decisions conditioned on all the received sensor data by generating, evaluating and updating multiple hypotheses as further observations are received.

Recent developments in multiframe 'track-before-detect' algorithms and multiple hypothesis tracking algorithms are converging towards a joint multiframe solution to target detection and tracking problems. In particular, consider the striking similarities between the recursive update equations for the dynamic programming approach to target detection (2.38) and the structured branching implementation of multiple hypothesis tracking (2.45). Both algorithms evaluate multiple hypotheses in terms of the likelihood that the hypothesis is valid conditioned on all the received observations. The update equation for the DPA has an additional term which reflects its first order Markov model for the target track. The primary difference between the two approaches lies in the evaluation of the likelihood ratio function. The DPA, like all 'track-before-detect' algorithms, evaluates a likelihood ratio which is a function of the assumed signal models for the target observations, while the SB-MHT algorithm evaluates a likelihood function which is a function of the probabilistic state space model for the target dynamics as a function of detected target observations. In the following chapter, a multiple hypothesis tracking implementation of Blostein and Huang's multistage hypothesis test algorithm will be presented as an attempt to unify these two paradigms. The resulting Multiple Multistage Hypothesis Test Tracking (MMHTT) algorithm provides a unified framework for the detection and tracking of low observable, point-source targets in digital image sequences.

Chapter 3

Multiple Hypothesis Sequential Detection and Tracking

A new approach for detecting and tracking point-source targets in a sequence of digital images will be developed in this chapter. The algorithm to be presented is a multiple hypothesis tracking algorithm which exploits a recent sequential detection algorithm, Multi-Stage Hypothesis Tests (MSHT) [40], for combined target detection and track initiation. The Multi-Stage Hypothesis Testing (MSHT) algorithm was originally proposed as a 'track-before-detect' solution to the detection of moving, sub-pixel targets [40]. However, there was no provision in the original implementation of the algorithm to extend the detected target trajectories and perform active tracking.

In this chapter an algorithm will be developed which exploits a new implementation of the MSHT algorithm in a multiple hypothesis tracking scheme. By incorporating the test statistic, from the sequential probability ratio test used in the MSHT algorithm, in a likelihood function for evaluating candidate track hypotheses a multiple hypothesis tracking algorithm is developed around the MSHT algorithm. The result is a new sequential algorithm for joint target detection and tracking. The new algorithm, Multiple Multistage Hypothesis Test Tracking (MMHTT), provides a unified framework for sequential detection and tracking in a multitarget environment.

Starting Assumptions

Reconsider the Multiframe Target Detection and Tracking Problem presented in Chapter 1. Given a sequence of K, $(M \times N)$ digital images

$$I[x,y,t] = \{I[x,y,t] \mid 1 \leq x \leq M,\, 1 \leq y \leq N,\, 1 \leq t \leq K\} \tag{3.1}$$

the MFTDT problem is to detect the presence of any targets in the image volume (figure 1.1) and to track the path of each detected target. The number of targets present and their initial positions and velocities is unknown. However, it will be assumed that the targets satisfy the point-source target model (2.2) of Chapter 2, the image sequence has been preprocessed to suppress background clutter, and the preprocessed image pixels can be modelled by (2.14) with

$$\begin{aligned} \mathbf{s_0} &= 0 \\ \mathbf{s_1} &= \mathbf{t} \end{aligned} \tag{3.2}$$

where the noise process $\mathbf{n}$ is assumed to be iid but not necessarily Gaussian. In general, the MMHTT algorithm can be applied to image sequences where the image pixels can be modelled as (2.12) with independent observations $\mathbf{y}$ such that

$$f_{\mathbf{Y}}(\mathbf{y} \mid \theta) = \prod_{y \in \mathbf{y}} f_{\mathbf{Y}}(y \mid \theta). \tag{3.3}$$

In fact, this model will be used in the feature point detection and tracking system described in Chapter 5.

3.1 Target Detection

The MMHTT algorithm employs a multiframe, time-domain 'velocity filter' (see section 2.2.5) approach to target detection. With this approach, target detection is treated as a multiple hypothesis detection problem. A large set of candidate target trajectories is constructed to test for the presence or absence of a target with an assumed trajectory. The candidate trajectory set is designed to span the range of expected target trajectories while maintaining an acceptable loss in detection performance due to trajectory mismatch.

The likelihood that a target with the assumed characteristics is present is then evaluated for each candidate trajectory. In the following, the multistage hypothesis test approach, [40], is used to evaluate the candidate trajectories. Undecided candidate trajectories are stored in a data structure which is indexed by the *current* hypothesized target location. This *local* tracking information is then used to implement a multiple hypothesis tracking scheme for the generation, evaluation and management of target track hypotheses for state estimation.

In this section the target detection functions of the MMHTT algorithm will be developed. The section begins with a discussion of the tree-structured candidate trajectory set and the multistage hypothesis test in [40]. Modifications and generalizations of Blostein and Huang's algorithm to facilitate multiple hypothesis tracking and improve computational efficiency will then be outlined and discussed. The key concepts are the use of a sequential decision procedure, with easily analyzable performance, for candidate trajectory evaluation and the imposition of a dynamic data structure to manage local detection and tracking information for multiple hypothesis tracking.

3.1.1 Forming candidate trajectories

In the following, it will be assumed that target trajectories can be approximated, at least locally, as linear, constant velocity trajectories. This is a valid assumption for many targets if the sensor frame rate is sufficiently high. The assumption of linear, constant velocity trajectories is not an inherent limitation of this approach, but it limits the computational demands on the resulting system. In general, candidate target trajectories may have arbitrary dynamics (e.g. curved, linear accelerating, etc.) but without some restrictions, the number of candidate trajectories is unbounded.

As discussed in section 2.2.5, if the target image can be approximated by a point on the detector array, then a discrete tree-structured candidate trajectory set can be designed to test a continuous range of target trajectories. Although any candidate trajectory set can be evaluated using the MMHTT algorithm, the discrete, tree-structured trajectory set is simple to analyze and highlights the reduction in computational complexity afforded by a sequential decision rule. Thus, the tree-structured candidate trajectory set will be developed in the following as an illustration of the benefits of the sequential approach.

If it can be assumed that the imaging system is ideal (i.e. the sensor optics preserve the impulse response of a point-source target) and the detector array is spatially contiguous (no gaps between detector elements), then the target image can be approximated by a point on the sensor focal plane. In practice, the sensor point-spread function has a finite blur radius. However, if the spatial resolution of the sensor optics is sufficiently great relative to the spatial resolution of the sensor detector array, then the target image response can be approximated by a point in the image plane. The resultant SNR loss can be calculated as a function of the target signal strength, the noise power and the sensor point-spread function.

For example, if a matched filter detector is used to test candidate trajectories for an ideally imaged target, with constant intensity A, in an additive iid Gaussian noise background, the SNR at the output of the matched filter is [63]

$$\begin{aligned} \mathrm{SNR}_{ideal} &= 10\log \tfrac{1}{\sigma^2} \textstyle\sum_{i=1}^{n} A^2 \text{ dB} \\ &= 10\log n + 10\log \tfrac{A^2}{\sigma^2} \text{ dB.} \end{aligned} \tag{3.4}$$

where $10\log\frac{A^2}{\sigma^2}$ is the intrinsic signal-to-noise ratio and $10\log n$ is the multiframe SNR gain. The output SNR of an equivalent matched filter for a target imaged by a real sensor, with a finite blur radius, can be expressed in terms of (2.11) as

$$\begin{aligned} \mathrm{SNR}_{opt} &= 10\log \frac{\sum_{\mathbf{y}} A^2 w_{j,k,l}^2}{\sigma^2} \text{ dB} \\ &= 10\log \textstyle\sum_{\mathbf{y}} w_{j,k,l}^2 + 10\log \tfrac{A^2}{\sigma^2} \text{ dB} \end{aligned} \tag{3.5}$$

The SNR loss incurred by assuming an ideal imaging system can then be expressed as a function of the mismatch between the correlator detector matched to an ideal point target and the true target shape and trajectory.

The output SNR of the mismatched filter is

$$\begin{aligned} \mathrm{SNR}_{subopt} &= 10\log \frac{\sum_{i=1}^{n} A^2 w_i}{\sigma^2} \\ &= 10\log \textstyle\sum_{i=1}^{n} w_i + 10\log \tfrac{A^2}{\sigma^2} \end{aligned} \tag{3.6}$$

where $w_i \equiv w_{0,0,i}$. Note that $\mathrm{SNR}_{ideal} \geq \mathrm{SNR}_{opt} \geq \mathrm{SNR}_{subopt}$. Thus, the SNR loss associated with the assumption of an ideal imaging system is no greater than

$$\mathrm{SNR}_{ideal} - \mathrm{SNR}_{subopt} = 10\log n - 10\log \sum_{i=1}^{n} w_i. \tag{3.7}$$

Thus, for a maximum SNR loss of 3 dB,

$$\sum_{i=1}^{n} w_i \geq n/2 \tag{3.8}$$

which implies that the assumption of an ideal imaging system is reasonable if on average half of the target intensity is received by pixels coincident with an ideal point target on the same trajectory.

In the following, it will be assumed that the system can be approximated as an ideal imaging system. The algorithm will be applied to a tree-structured candidate trajectory set, although in general, the algorithm could be applied to any candidate trajectory set. The choice of a candidate trajectory set is a trade-off between the loss in detection performance due to trajectory mismatch and the computational cost of additional candidate trajectories.

Constructing a Tree-Structured Candidate Trajectory Set

In this section, an algorithm will be described for constructing a complete tree-structured candidate trajectory set for a point-source target with a known range of velocities. Typical MTT systems are designed to detect and track a particular class of real objects. The physical constraints of target dynamics and sensor imaging characteristics bound the target's perceived velocity on the sensor focal plane. Thus, it is often reasonable to assume a known target velocity range.

Figure 3.1: Target Velocity Annulus

As discussed in section 2.2.5, under the assumption of an ideal imaging system, a dense hypothesis tree can be constructed to evaluate all of the linear, constant velocity point-source target trajectories in an image sequence volume $I[x, y, t]$. Consider the annulus in figure 3.1 centred on the detector pixel (x_j, y_k). A candidate trajectory could originate from any point (x_0, y_0) on the surface of the detector pixel

$$(x_0, y_0) \ni (j-1)\Delta x \leq x_0 \leq j\Delta x,\ (k-1)\Delta y \leq y_0 \leq k\Delta y. \tag{3.9}$$

Given the target's initial position (x_0, y_0) and a known velocity range, the target's position in subsequent frames is constrained to lie within the annulus centred at (x_0, y_0) with inner radius $r_{min} = v_{min}(n\Delta t)$ and outer radius $r_{max} = v_{max}(n\Delta t)$. If the centre of pixel (x_j, y_k) is assigned relative offset coordinates $(0, 0)$, then any pixel with relative offset coordinates $(x_{n,j}, y_{n,k})$ which satisfy one of

$$\begin{aligned}
v_{min}(n\Delta t) &\leq \sqrt{(x_{n,j}+1/2)^2 + (y_{n,k}+1/2)^2} \leq v_{max}(n\Delta t)\\
v_{min}(n\Delta t) &\leq \sqrt{(x_{n,j}+1/2)^2 + (y_{n,k}-1/2)^2} \leq v_{max}(n\Delta t)\\
v_{min}(n\Delta t) &\leq \sqrt{(x_{n,j}-1/2)^2 + (y_{n,k}+1/2)^2} \leq v_{max}(n\Delta t)\\
v_{min}(n\Delta t) &\leq \sqrt{(x_{n,j}-1/2)^2 + (y_{n,k}-1/2)^2} \leq v_{max}(n\Delta t)
\end{aligned} \tag{3.10}$$

in the n'th subsequent frame, is considered in range.

The construction of a candidate trajectory tree can be likened to the construction of a tree of discrete trajectories spanning the pixels in the image volume enclosed by the expanding velocity annulus. The problem is to find the sequences of pixels in consecutive image frames which are consistent with a linear, constant velocity trajectory originating in the root pixel (x_j, y_k).

The first node of the tree and hence the first observation for all candidate trajectories is $I[x_j, y_k, t]$. After the subsequent frame, the candidate trajectory set consists of the set of two pixel paths from (x_j, y_k) to the first tier of pixels $(x_{1,j}, y_{1,k})$ satisfying (3.10) for $n = 1$. Note that each trajectory shares the same first observation and that a new tree node is constructed for each pixel in the second tier.

Subsequent tiers of pixels are added recursively. First, the annulus is expanded to define the set of potential target pixels in the new tier and a new tree node is constructed for each pixel in the tier. Then each node is evaluated as a potential extension of each node from the previous tier.

The following conditions must be satisfied for a pixel to be accepted as a valid extension of a current trajectory:

1. the new pixel must be further from the origin $(0, 0)$ than its parent,
2. the centre-to-centre distance between the pixels must be traversable in a single frame by a target with a velocity $v \ni v_{min} \leq v \leq v_{max}$,
3. it must be possible to draw a straight-line through all the pixels in the current candidate trajectory which intersects the new pixel, and
4. the average velocity of the current trajectory must be within the range of velocities that could cover the old → new pixel distance in a single frame.

Algorithm 1 (Construction of a K stage candidate trajectory tree)

```
Let $(x_{0,j}, y_{0,k}) = (x_j, y_k)$ be the first tree node in tier 0
For subsequent image frames n := 1 to K-1
  For each $(x_{n,j}, y_{n,k})$ which satisifies (3.10)
    For each node in tier n-1
      Let $d = \|(x_{n,j} - x_{n-1,j}, y_{n,k} - y_{n-1,k})\|$
      If $0 \leq v_{min} \leq d \leq v_{max}$ and $d - \sqrt{2} < \frac{1}{n-1}\|(x_{n-1,j}, y_{n-1,k})\| < d + \sqrt{2}$
      and if $(x_{0,j}, y_{0,k})$, $(x_{n-1,j}, y_{n-1,k})$, $(x_{n,j}, y_{n,k})$ are colinear
      then add a path from the tier n-1 node to a new node at $(x_{n,j}, y_{n,k})$ in tier n
The candidate trajectory set is the set of all paths from $(x_j, y_k)$ to pixels in tier K-1.
```

Storing Candidate Trajectories

Since many trajectories share common observations in the first few test stages the candidate trajectories are stored in a hierarchical lookup-table. Each entry in the table contains:

1. the current test stage,
2. the relative offset from the root pixel, and
3. a list of pointers to the table entries for valid extensions of the current trajectory in the next test stage.

Thus, as the test progresses and the candidate trajectories diverge, the number of entries in the table for a single test stage increases. Typically, for a known range of target velocities (speed and direction), the lookup-table for the candidate trajectory set can be constructed off-line. Thus, while the MTT system is on-line, the lookup-table stores all the information necessary to propagate the multistage hypothesis tests for each candidate trajectory.

As an example of this process, consider the construction of a candidate trajectory tree for linear, constant velocity targets with a velocity range of $0 \rightarrow 1$ pixels/frame. Define the root node as a relative displacement of $(0, 0)$. The velocity annulus for this problem has an inner radius of 0 and an outer radius of n pixels after n frames (see figure 3.2). Each numbered pixel in figure 2.1 represents a node of the candidate trajectory hypothesis tree. The figure illustrates four of the nine possible trajectories that pass through node 1 and node 2 in the first two test stages. The corresponding lookup-table entries are illustrated in figure 3.3.

Figure 3.2: Tree-Structured Trajectory Search

Note that the regular tesselation of the image pixels introduces symmetries in the candidate trajectory tree which can be exploited to reduce the memory storage requirements of the candidate trajectory lookup-table at a slight cost of increased overhead in calculating the relative offsets. For example, the sets of candidate trajectories passing through the *corner* pixels $\{(1,1), (-1,-1), (-1,1), (1,-1)\}$ in the second stage have the same relative structure and could be represented by a single set of entries in the lookup-table. For the current example, the 9 subtrees covering the diversity of candidate trajectories in the second test stage could be represented by lookup-table entries for 3 complete trees (i.e. 1 for 4 corner subtrees, 1 for 4 cross subtrees and a centre subtree). Relative offsets for the remaining 6 subtrees could be computed by interchanging the x and y coordinates or changing the sign of the offset coordinates as required.

A full tree to cover this range of velocities for 10 consecutive image frames (a 10-stage tree) has 194 387 nodes. All 194 387 candidate trajectories share the same first observation, an average of 21 598 share the same second observation and an average of 3 967 have their first three observations in common. Clearly, a decision to accept a test hypothesis at an early stage greatly reduces the number of candidate trajectories that need to be evaluated. Thus, the sequential detection approach in [40] will be employed to evaluate the candidate trajectories.

3.1.2 Track Confirmation

Given a set of candidate trajectories for target detection, the next task is to test each trajectory in the set, for the presence of a target. Each test is posed as a binary hypothesis test (2.14) for the observations $\mathbf{y}$ in the candidate trajectory set. It is assumed in the following that the observations are mutually independent, and identically distributed. In practice, however, the observations are not always identically distributed. Targets with similar trajectories or intersecting trajectories may cause a candidate trajectory to contain a mixture of observations with different probability distributions. This leads to an increase in the detection error rate which will be analyzed in Chapter 4.

Following [40], a sequential decision rule is applied to the binary hypothesis tests for each candidate trajectory. A sequential decision rule is preferable to a batch or fixed sample size (FSS) rule because it is data adaptive. A FSS rule makes the best decision after receiving all n observations while a sequential rule, designed for the same detection error probabilities, evaluates the observations as they are received, making a decision as soon as the specified error probabilities can be satisfied.

It can be shown that, on average, a sequential probability ratio test requires fewer samples than a FSS test to meet the same detection performance specifications [83, 63]. This is a particularly desirable property for a MTT system. It is clear that rapid target evaluation is an intrinsically important property of a MTT system; however, the results of the detection decision can also be used to reduce the computational load by aiding in track initiation and evaluation decisions (see section 3.3.1).

As in [40], a truncated SPRT will be used to test the candidate trajectories. The truncated SPRT can be viewed as a trade-off between an SPRT with constant thresholds $\hat{a}$ and $\hat{b}$, and an FSS test with fixed threshold τ and fixed sample size K. In particular, a K stage truncated SPRT is a multistage hypothesis test with constant thresholds $a_i = \hat{a}$ and $b_i = \hat{b}$, where the decision rule for stages $1 \leq i \leq K-1$ is

$$\sum_{j=1}^{i} z_j \begin{cases} \geq \hat{a} & \Rightarrow \text{choose } H_1 \\ \leq \hat{b} & \Rightarrow \text{choose } H_0 \\ \in (\hat{b}, \hat{a}) & \Rightarrow \text{take another sample} \end{cases} \tag{3.11}$$

and the decision rule after K stages is,

$$\sum_{j=1}^{K} z_j \begin{cases} \geq \tau & \Rightarrow \text{choose } H_1 \\ < \tau & \Rightarrow \text{choose } H_0 \end{cases} \tag{3.12}$$

for an observed realization z_j of the random variable

$$Z_j = \ln L(y_j). \tag{3.13}$$

The detection error probabilities for the truncated SPRT are

$$\begin{aligned} \alpha &= \alpha_{SPRT} + \alpha_{FSS} \\ \beta &= \beta_{SPRT} + \beta_{FSS} \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} \alpha_{FSS} &= c_0\,\alpha \\ \alpha_{SPRT} &= (1-c_0)\,\alpha \\ \beta_{FSS} &= c_1\,\beta \\ \beta_{SPRT} &= (1-c_1)\,\beta \end{aligned} \tag{3.15}$$

and the weights c_0 and c_1 control the behaviour of the test. If c_0 and c_1 are both zero then the resulting test is an SPRT, and if c_0 and c_1 are both equal to one then the resulting test is a FSS test. Thus, a truncated SPRT can be designed to behave like an SPRT, with a small average sample size under H_0 or H_1 that is sensitive to parameter mismatch, or like an FSS test, with a higher average sample size but less sensitivity to parameter mismatch. It should be noted that a truncated SPRT is **not** equivalent to performing an SPRT for the first $K-1$ stages followed by an optimal FSS test in the K'th stage. The truncated SPRT assigns a proportion of its accepted error probabilities to each of its sequential and fixed decisions.

Note that the error probabilities α and β are nominal values used to design the thresholds

$$\begin{aligned} \hat{a} &= \ln\left[\tfrac{1-(1-c_1)\beta}{(1-c_0)\alpha}\right] \\ \hat{b} &= \ln\left[\tfrac{(1-c_1)\beta}{1-(1-c_0)\alpha}\right] \end{aligned} \tag{3.16}$$

In practice, the realized detection error probabilities are somewhat less than the nominal design values [83].

The truncation stage K and the fixed threshold τ are chosen to satisfy

$$\begin{aligned} \Pr(\textstyle\sum_{j=1}^{K} z_j \geq \tau \mid \mathbf{Y} \sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \theta_{H_0})) &= \alpha_{FSS} \\ \Pr(\textstyle\sum_{j=1}^{K} z_j < \tau \mid \mathbf{Y} \sim f_{\mathbf{Y}}(\mathbf{y} \mid \theta \in \theta_{H_1})) &= \beta_{FSS} \end{aligned} \tag{3.17}$$

which for Gaussian probability density functions $f_Y(y \mid \theta \in \theta_{H_0}) = N(\mu_0, \sigma^2)$ and $f_Y(y \mid \theta \in \theta_{H_1}) = N(\mu_1, \sigma^2)$, reduces to

$$K = \left[\Phi^{-1}(c_0\,\alpha) + \Phi^{-1}(c_1\,\beta)\right]^2 \left(\frac{\sigma}{\mu_1 - \mu_0}\right)^2 \tag{3.18}$$

and

$$\tau = \sqrt{K}[\mu_1\Phi^{-1}(c_0\,\alpha) - \mu_0\Phi^{-1}(c_1\,\beta)] \left(\frac{\sigma}{\mu_0 - \mu_1}\right)^2 . \tag{3.19}$$

where Φ is the standard (normalized) Gaussian distribution function $N(0,1)$ and Φ^{-1} is its inverse.

Keeping Track of Undecided Trajectories

Due to the sequential nature of the track confirmation algorithm, detection decisions may be postponed for several stages. Consequently, the MMHTT algorithm needs a mechanism for storing the information associated with undecided trajectories. The data structure used to store this information is depicted in figure 3.3.

In contrast to the original formulation of the MSHT algorithm [40], which stored undecided trajectories relative to their root pixel location, the undecided trajectories are stored relative to their current image location. This provides a natural clustering of the undecided trajectories which is valuable in making track initiation and track maintenance decisions (see sections 3.2.5 and 3.3.1).

Figure 3.3: The Undecided Trajectory Data Structure

Each pixel location has an associated undecided trajectory list, which is a linked list of candidate trajectory records containing the following fields:

1. a pointer to the next field in the list,
2. the identifier for the current node in the hypothesis tree,
3. the current value of the likelihood ratio test statistic,
4. a list of the last N observations, and
5. any auxiliary data which may be required for a given hypothesis test (e.g. current sample mean or variance).

The hypothesis node identifier is an index to the lookup-table storing the candidate trajectories. Each entry in the lookup-table contains a pixel offset from the trajectory's point of origin, the current trajectory test stage and the possible trajectory extensions in the next stage. By computing the candidate trajectory lookup-table offline, the algorithm can efficiently propagate candidate trajectory tests by accessing the lookup-table with its node identifier index.

Algorithm 2 (Modified Multistage Hypothesis Test Algorithm)

1. *Construct hierarchical lookup-table for candidate trajectory set.*
2. *Construct hypothesis test threshold lookup-table indexed by test stage.*
3. *Initialize empty undecided trajectory lists for every image pixel location.*
4. *For each image in the preprocessed image sequence, update the undecided trajectory lists for each pixel location (x, y), by propagating the current test to each child of the current node,*
   (a) *Retrieve the relative pixel offset of each child node from the lookup-table.*
   (b) *Calculate the child node location $\mathbf{x}_{child} = offset_{new} - offset_{old} + (x, y)$.*
   (c) *Use $I[\mathbf{x}_{child}]$ to update the multistage hypothesis test.*
   (d) *If H_0 is accepted, delete the child entry.*
   (e) *If H_1 is accepted, pass the child entry to the tracking algorithm.*
   (f) *Otherwise, add the child entry to the list at $\mathbf{x}_{child}$*
5. *Initiate new multistage tests (see section 3.3.1).*

3.1.3 Summary

This section has described the target detection phase of the MMHTT algorithm. For each pixel where a target search is initiated, a set of candidate trajectories is constructed to evaluate all the linear, constant velocity trajectories originating in the root pixel and meeting the specified bounds on target velocity. Each candidate trajectory is tested for the presence/absence of a target with a truncated Sequential Probability Ratio Test designed to meet the specified system detection error probabilities (see Chapter 4). The information associated with undecided trajectories is stored in a dynamic data structure which is indexed to the current image pixel location of the candidate trajectory. Thus, the detection phase converts the preprocessed image signal into a list of confirmed target trajectory segments with estimates of target position, velocity and a measure of confidence in the detection decision. The key difference between the current approach and Blostein's original implementation of the MSHT algorithm is the local indexing of the undecided trajectory data structure. This modification will be exploited in the following section to implement a multiple hypothesis tracking scheme to manage and extend the confirmed target trajectories generated by the target detection phase.

3.2 Target Tracking

The sequential detection procedure presented in the preceding section was originally proposed as a 'track-before-detect' solution to the problem of detecting moving, subpixel targets in a sequence of digital images [40]. However, there was no provision in the original implementation to extend the detected target trajectories and actively track the detected targets. In this section, the modified implementation of the MSHT algorithm, presented in the previous section, will be exploited to fulfill the track initiation requirements of a multitarget tracking system.

As discussed in section 2.3, the function of a multitarget tracking system is to identify individual targets in the sensor's field-of-view and to estimate the *state* of each target from the detected target observations. This is accomplished by partitioning the detected target observations into disjoint sets, or target tracks, and estimating the *state* of each target from the observations in the target's track. The new algorithm to be presented in this section extends the target trajectory segments detected by the MSHT algorithm to form extended target tracks suitable for state estimation and prediction.

The algorithm, Multiple Multistage Hypothesis Test Tracking, is a 'track-oriented' implementation of multiple hypothesis tracking (see section 2.3.2) that exploits the MSHT algorithm for track initiation and confirmation. This approach is conceptually similar to the Structured Branching implementation of multiple hypothesis tracking proposed by Blackman et al. [35]. Both algorithms evaluate candidate target tracks with a sequential probability ratio test. However, the likelihood function employed by the SB-MHT algorithm (2.46) is derived from a state space model of the target dynamics while the likelihood function employed by the proposed algorithm (2.15) is derived from a model of the sensor signal in the presence/absence of a target.

These two approaches are philosophically different. The state space approach emphasizes the position of the candidate target observations relative to the position predicted by the state space model of the target dynamics while the current approach emphasizes the signal characteristics of the received observations relative to the expected signal models. These differences are motivated by the distinct heritage of the two approaches. The SB-MHT algorithm is a product of the target tracking community which emphasizes state space models of the target dynamics and the current algorithm is an extension of a signal detection algorithm which emphasizes the sensor signal models. In principle, these approaches could be combined by including the signal information as an additional state and modifing the likelihood function appropriately.

3.2.1 The Detected Target Observations

The target detection algorithm presented in the preceding section makes detection decisions, for the candidate target trajectories, over multiple image frames. A decision to accept the target present hypothesis implies an acceptance of the hypothesis that a target is present in each image observation along the candidate trajectory. Thus, the output of the detection algorithm consists of sets of detected target observations along short segments of the detected target's trajectory.

When the k'th multistage hypothesis test terminates, after i frames, the candidate trajectory data structure contains the position of the target in the current image frame, the position of the target in the frame in which the test was initiated, the current test stage and the likelihood ratio test statistic

$$\begin{aligned} L_{t_k} &= \textstyle\sum_{j=1}^{i} z_j \\ &= \textstyle\sum_{j=1}^{i} \ln L(y_j) \\ &= \ln \textstyle\prod_{j=1}^{i} \frac{\Pr(y_j|H_1)}{\Pr(y_j|H_0)} \\ &= \ln \frac{\Pr(y_1,\ldots,y_j|H_1)}{\Pr(y_1,\ldots,y_j|H_0)}. \end{aligned} \tag{3.20}$$

For an assumed linear, constant velocity trajectory, the candidate trajectory endpoints and the total length of the test are sufficient to estimate the velocity of the target and its position in every intermediate test stage (frame). In addition, the test statistic L_{t_k} provides a measure of the reliability of the target detection decision under the assumed data hypotheses.

The probability that the target present hypothesis is valid, conditioned on the image observations $\mathbf{Y}_{t_k} = \{y_1, \ldots, y_i\}$, can be expressed in terms of the log-likelihood ratio test statistic L_{t_k}:

$$\begin{aligned}
\Pr(H_1 \mid \mathbf{Y}_{t_k}) &= \frac{\Pr(\mathbf{Y}_{t_k} \mid H_1)\Pr(H_1)}{\Pr(\mathbf{Y}_{t_k})} \\
&= \frac{\Pr(\mathbf{Y}_{t_k} \mid H_1)\Pr(H_1)}{\Pr(\mathbf{Y}_{t_k} \mid H_1)\Pr(H_1) + \Pr(\mathbf{Y}_{t_k} \mid H_0)\Pr(H_0)} \\
&= \frac{\dfrac{\Pr(\mathbf{Y}_{t_k} \mid H_1)\Pr(H_1)}{\Pr(\mathbf{Y}_{t_k} \mid H_0)\Pr(H_0)}}{1 + \dfrac{\Pr(\mathbf{Y}_{t_k} \mid H_1)\Pr(H_1)}{\Pr(\mathbf{Y}_{t_k} \mid H_0)\Pr(H_0)}} \\
&= \frac{\exp(L_{t_k} + C_{t_k})}{1 + \exp(L_{t_k} + C_{t_k}),}
\end{aligned} \tag{3.21}$$

where

$$C_{t_k} = \ln\left[\frac{\Pr(H_1)}{\Pr(H_0)}\right] \tag{3.22}$$

is the log-ratio of the prior probability that the image observations $\mathbf{Y}_{t_k}$ are or are not coincident with a target trajectory. This prior probability is a measure of the expected target density and reflects the proportion of target to non-target candidate trajectories.

Since H_1 and H_0 are mutually exclusive hypotheses which partition the sample space of the image observations $\mathbf{Y}_{t_k}$, it follows that

$$\begin{aligned}
\Pr(H_1|\mathbf{Y}_{t_k}) &= \tfrac{\exp(L_{t_k}+C_{t_k})}{1+\exp(L_{t_k}+C_{t_k})}, \text{ and} \\
\Pr(H_0|\mathbf{Y}_{t_k}) &= \tfrac{1}{1+\exp(L_{t_k}+C_{t_k})}.
\end{aligned} \tag{3.23}$$

Thus, the MSHT algorithm detects the target and identifies its position and velocity over several image frames, satisfying the classical multitarget tracking functions of track initiation and track confirmation. The only remaining task to satisfy the data association requirements of a complete MTT system is to extend and maintain the target tracks initiated by the detection algorithm.

The standard data association algorithms in Chapter 2 were designed to associate discrete target detection observations in each image frame. It is unclear how these algorithms could be applied to associate the sets of observations associated with detected target trajectory segments over multiple frames of the image sequence. However, as the following section will discuss, the sequential decision process presented in the preceding section is compatible with a track-oriented multiple hypothesis tracking scheme.

3.2.2 Multiple Hypothesis Tracking

As discussed in section 2.3.2, multiple hypothesis tracking algorithms are multiframe Bayesian probabilistic algorithms which attempt to consider all possible data association hypotheses over a number of frames. Each hypothesis consists of a set of observation-to-track associations, where each detected target observation is either

1. classified as a false alarm,
2. associated with one of a number of hypothesized target tracks, or
3. classified as a new target.

The multiple hypothesis tracking approach is to maintain a number of probable data association hypotheses with the foreknowledge that the most probable hypothesis at any given stage may be the continuation of a less probable hypothesis from a previous stage. By maintaining and evaluating multiple data association hypotheses these algorithms have an ability to postpone difficult association decisions pending the receipt of new observations. The ability to use later observations to resolve prior association decisions greatly reduces the number of association errors, and results in a substantial performance gain relative to single frame algorithms.

There are two major components to a multiple hypothesis tracking algorithm. The first component is an efficient technique for generating and managing the multitude of association hypotheses and the second component is a technique for evaluating competing hypotheses. The discussion of hypothesis generation and management will be deferred until section 3.2.4 while some of the issues associated with hypothesis evaluation are explored.

In a standard implementation of multiple hypothesis tracking, competing hypotheses are evaluated by calculating the *a posteriori* probability, $\Pr(H_l \mid D_k)$, (see (2.40)) that each hypothesis H_l is valid, conditioned on the detected target observations. If the observations are independent, then this probability can be evaluated recursively in terms of the most recent observation update. Note that this recursive relationship requires an evaluation of all the association hypotheses for the most recent observation update (2.42).

In practice, many if not most of these hypotheses are highly improbable. Thus, as the number of detected target observations increases, the cost of evaluating a large number of improbable hypotheses grows rapidly and this approach becomes extremely inefficient in its use of computational resources. Consequently, Blackman et al. proposed a 'track-oriented' implementation of multiple hypothesis tracking which replaces the evaluation of global hypotheses (2.40) with an evaluation of individual track hypotheses (2.43) from which the $\Pr(H_l \mid D_k)$ may be computed [35].

The Structured Branching implementation of multiple hypothesis tracking, evaluates each target track with a sequential probability ratio test designed to test the observations against the binary hypotheses

- H_1: target present, $Y_i \sim f(y_i \mid H_1)$ and
- H_0: target absent, $Y_i \sim f(y_i \mid H_0)$ (all observations are false alarms).

This is conceptually identical to the evaluation of candidate trajectories by the MSHT algorithm. However, the SB-MHT algorithm evaluates each candidate trajectory with respect to a probabilistic model of the target's dynamics (2.46).

Those tracks which are accepted by the SPRT as target tracks are propagated and used to generate global hypotheses. Each global hypothesis consists of a set of track hypotheses, where a given global hypothesis is valid, if and only if all of its constituent track hypotheses are also valid and the remaining detected target observations are false alarms. Thus, global hypotheses can be evaluated indirectly by evaluating the validity of their component track hypotheses.

Since each track is a member of more than one global hypothesis, there are substantially fewer tracks than global hypotheses. Thus, by eliminating improbable tracks prior to the evaluation of global association hypotheses the computational cost of maintaining multiple hypotheses is substantially reduced. In the following, a similar conceptual structure will be exploited to extend the MSHT algorithm in a multiple hypothesis tracking framework.

3.2.3 Track Hypotheses

In the following, the approach used to develop the SB-MHT algorithm, will be used to develop a 'track-oriented' implementation of mulitple hypothesis tracking based on the MSHT detection algorithm. As in [35], the validity of each candidate trajectory t_k is measured by a likelihood ratio test statistic, L_{t_k} (recall (3.20)), in this case the likelihood statistic used in the sequential target detection decision. Thus, the likelihood of each candidate trajectory is initially evaluated by the MSHT algorithm. If the detection test terminates and the target absent hypothesis H_0 is accepted, then the candidate trajectory is removed from further consideration. If however, the detection test terminates and the target present hypothesis H_1 is accepted, an explicit means to extend and track the detected target trajectory is required.

A straightforward approach would be to extend the current candidate trajectory along its current linear, constant velocity trajectory. However, it was initially assumed that the target trajectory could be approximated by a linear, constant velocity trajectory over a short time interval (see section 3.1.1). In practice, the extended target trajectory has a finite curvature and the target dynamics may vary if the target manoeuvres. Thus, a target tracking algorithm must be able to follow the target trajectory and detect target manoeuvres over an extended period of time, relying only on local linearity of the target trajectory.

The solution proposed herein is to reinitiate a candidate trajectory search in every pixel where the multistage hypothesis test terminates and a target is declared present. Although the initial candidate trajectory search was constrained only by the expected range of target velocities (speed and direction), the search for the extension of a detected target trajectory can be further constrained by a maximum allowable change in target velocity, including target manoeuvers. This additional constraint exploits the smoothness of the target trajectory and prior knowledge of the target's manoeuvring abilities to reduce the effective trajectory search space without sacrificing tracking performance. Thus, the scope of the candidate trajectory search for a confirmed target trajectory is constrained around an estimate of the nominal target velocity implied by the initial detected trajectory segment. In principle, this approach can be refined by appealing to the error covariance estimate of the target state estimator to modulate the scope of the constrained search.

The search for extensions of a confirmed candidate trajectory consists of an evaluation of the candidate trajectories in the candidate trajectory set coincident with the pixel in which the candidate trajectory was confirmed. However, as will become evident in the following discussion, for the purposes of track hypothesis evaluation it is desirable to maintain the statistical independence of the candidate trajectory segments which constitute a hypothesized target track. Thus, hypothesized target tracks consisting of disjoint candidate trajectory segments are formed by evaluating, as extensions of the current track, trajectories formed by ignoring the root pixel of the candidate trajectories in the candidate trajectory set which satisfy the constraints on the maximum allowable change in target velocity.

With the proposed approach, candidate trajectories can be extended indefinitely by exploiting multiple multistage hypothesis tests to detect short, linear segments of the detected target's trajectory. As the candidate trajectories are propagated, multiple track hypotheses are formed to describe the multitude of ways in which the detected trajectory segments could be associated to form candidate target tracks. Thus, this new approach is termed Multiple Multistage Hypothesis Test Tracking (MMHTT).

Defining the observations along the k^{th} hypothesized target track T_k as the concatenation of the observations of its constituent candidate trajectory segments

$$\mathbf{Y}_{T_k} = \bigcup_{t_i \in T_k} \mathbf{Y}_{t_i} \tag{3.24}$$

where

$$\mathbf{Y}_{t_i} \bigcap \mathbf{Y}_{t_j} = \emptyset,\ \forall (t_i, t_j \in T_k \mid i \neq j) \tag{3.25}$$

the probability that T_k represents a true target track can be expressed as

$$\Pr(T_k) = \Pr(\bigcap_{t \in T_k} \mathbf{Y}_t \sim f_Y(\mathbf{y} \mid \theta \in \theta_{H_1})) \tag{3.26}$$

Expressing $\mathbf{Y}_{T_k}$ as an ordered set of n candidate trajectory segments $\mathbf{Y}_{t_i}$,

$$\mathbf{Y}_{T_k} = \{\mathbf{Y}_{t_1}, \ldots \mathbf{Y}_{t_n}\} \tag{3.27}$$

one can evaluate the probability of concurrent, colinear track hypotheses

$$\{T_j \mid \mathbf{Y}_{T_j} \subseteq \mathbf{Y}_{T_k}\} \tag{3.28}$$

as

$$\begin{array}{l} \Pr(T_j) = \Pr(\,(\bigcap_{t_i \in T_j} \mathbf{Y}_{t_i} \sim f_Y(\mathbf{y} \mid \theta \in \theta_{H_1}))\bigcap(\bigcap_{t_i \notin T_j} \mathbf{Y}_{t_i} \sim f_Y(\mathbf{y} \mid \theta \in \theta_{H_0}))) \\ \Pr(T_\emptyset) = \Pr(\bigcap_{t_i \in T_k} \mathbf{Y}_{t_i} \sim f_Y(\mathbf{y} \mid \theta \in \theta_{H_0})). \end{array} \tag{3.29}$$

A local evaluation of the probability that T_k is valid, conditioned on the detected target observations, $\mathbf{Y}_{T_k}$, can then be evaluated by accumulating the test statistics L_{t_i} of its constitutent trajectory segments via

$$L_{T_k} = \sum_{t_i \in T_k} L_{t_i}. \tag{3.30}$$

Using a similar argument to (3.21),

$$\begin{aligned} \Pr(T_k \mid \mathbf{Y}_{T_k}) &= \frac{\Pr(\mathbf{Y}_{T_k} \mid T_k)\Pr(T_k)}{\Pr(\mathbf{Y}_{T_k})} \\ &= \frac{\dfrac{\Pr(\mathbf{Y}_{T_k} \mid T_k)\Pr(T_k)}{\Pr(\mathbf{Y}_{T_k} \mid T_\emptyset)\Pr(T_\emptyset)}}{1 + \sum_{T_j} \dfrac{\Pr(\mathbf{Y}_{T_k} \mid T_j)\Pr(T_j)}{\Pr(\mathbf{Y}_{T_k} \mid T_\emptyset)\Pr(T_\emptyset)}}\,, \\ &= \frac{\exp(L_{T_k} + C_{T_k})}{1 + \sum_{T_j} \exp(L_{T_j} + C_{T_j})} \end{aligned} \tag{3.31}$$

where

$$C_{T_j} = \ln\left[\frac{\Pr(T_j)}{\Pr(T_\emptyset)}\right] \tag{3.32}$$

is the log-ratio of the prior probability that the image observations $\mathbf{Y}_{T_j}$ are or are not coincident with a target trajectory. The priors for the track hypotheses can, for example, be derived from a statistical model of the expected track length [7]. It is assumed that the observations $\mathbf{Y}_{T_k}$ are described by one of the mutually exclusive track hypotheses T_j or $T_\emptyset$. Therefore, it follows that

$$\begin{aligned}\Pr(T_k|\mathbf{Y}_{T_k}) &= \frac{\exp(L_{T_k}+C_{T_k})}{1+\sum_{T_j}\exp(L_{T_j}+C_{T_j})}, \text{ and}\\ \Pr(T_\emptyset|\mathbf{Y}_{T_k}) &= \frac{1}{1+\sum_{T_j}\exp(L_{T_j}+C_{T_j})}.\end{aligned} \tag{3.33}$$

Note that the ability to express the track hypothesis score L_{T_k} as the sum of the scores of its constituent candidate trajectory segments is critically dependent on the statistical independence of the candidate trajectories. Since the image observations are assumed to be mutually independent, defining the hypothesized target track $\mathbf{Y}_{T_k}$ as a set of disjoint but connected sets of image observations $\mathbf{Y}_{t_i}$ is sufficient to enable $\Pr(\mathbf{Y}_{T_k} \mid T_j)$ to be expressed as

$$\Pr(\mathbf{Y}_{T_k} \mid T_k) = \prod_{i=1}^{n} \Pr(\mathbf{Y}_{t_i} \mid T_k). \tag{3.34}$$

Hence, the image observations for candidate extensions of detected target trajectory segments do not include the image observation at the root of the candidate trajectory set.

Summarizing, the MMHTT algorithm hypothesizes candidate target tracks from the trajectory segments detected by the MSHT algorithm. Each target track consists of a unique sequence of candidate trajectory segments linking test initiation and confirmation points in multiple image frames. The set of all target tracks represents an inconsistent set of local observation-to-track associations in the sense that each track is a local association of observations to a particular candidate target track, but certain pairs of tracks represent mutually exclusive hypotheses. Associated with each track is a confidence measure or track score L_{T_k} and a set of detected target observations suitable for target state estimation.

3.2.4 Global Hypotheses

For typical multistage hypothesis tests, there is a small but finite probability that a single multistage test will accept the target present hypothesis in the absence of a target. However, in dense, multitarget environments, false target trajectories commonly arise as the result of candidate trajectories *stealing* target observations from several different true target trajectories. Since a target is present in each image observation, the probability that such a track is valid will be high based on an evaluation of the track hypotheses. Thus, an evaluation of the global data association problem is required to correctly resolve individual target tracks in a multitarget scenario.

Global hypotheses are formed from sets of compatible track hypotheses. A common assumption is that no more than one target may be present in a single image observation. Thus, two tracks are considered compatible if they have no observations in common. If more than one target is present in a single image observation or if a target can split into multiple targets, then this assumption is not well-founded and group tracking techniques should be applied (see, for example, Blackman [7]).

It will also be assumed that the candidate trajectories in a new trajectory search represent new target hypotheses while the candidate trajectories in a reinitiated search represent extensions of old target hypotheses. Note that when a candidate trajectory is accepted as the extension of an existing track, a new track hypothesis is formed which has the the root pixel of the parent and the combined observations of the original track and its accepted extension. The original track hypothesis is maintained to generate alternate extensions and to represent the hypothesis that the original track terminates (loss of track).

With standard single frame association algorithms, the most probable global hypothesis would be accepted after every frame and alternate hypotheses deleted. However, the multiple hypothesis tracking approach adopted herein allows the use of future observations to resolve prior association decisions. The resultant reduction in correspondence errors is well documented [7, 35] but the necessity to maintain and evaluate multiple alternate hypotheses is often criticized for its computational cost.

As discussed previously, the evaluation of the probability that a given global association hypothesis is valid requires an evaluation of all related hypotheses. The number of hypotheses to be evaluated is greatly reduced by using the MSHT algorithm to evaluate and confirm track hypotheses prior to the generation of global hypotheses. However, a further reduction in the number of hypotheses to be evaluated can be achieved by clustering the track hypotheses into sets of non-interacting (independent) track hypotheses and evaluating the global hypotheses for each cluster independently. This approach was first proposed by Reid [11] and is commonly used by MHT algorithms to divide the global association problem into smaller independent problems [7, 35].

Clustering Hypotheses

A track hypothesis cluster is defined as a minimal set of interacting tracks. Incompatible tracks are considered to interact directly but interaction is defined as a transitive operation, so if track T_a interacts with track T_b, and track T_b interacts with track T_c, then track T_a interacts with track T_c. Thus, Tracks T_a and T_c are considered to interact indirectly and all three tracks T_a, T_b, and T_c are members of the same cluster.

Global hypotheses can then be formed from sets of compatible tracks within a cluster. Each set of tracks represents one hypothesis regarding the true origin of the observations within the cluster. The total number of hypotheses that can be generated from a single cluster depends on the interactions between the track hypotheses (direct or indirect) but increases rapidly with the number of tracks in the cluster. Thus, from a computational viewpoint, it is desirable to minimize the size of individual clusters.

Evaluating Global Hypotheses

Following the approach in [35], the probability that a given hypothesis, H_j, is valid can be evaluated as a simple function of the scores of its constituent track hypotheses.

$$L_{H_j} = \sum_{T_i \in H_j} L_{T_i} + C_{T_i} \tag{3.35}$$

Define the observations in the k^{th} track hypothesis cluster C_k as

$$\mathbf{D_k} = \bigcup_{T_i \in C_k} \mathbf{Y}_{T_i} \tag{3.36}$$

and hypotheses H_j and $H_\emptyset$ as

$$\begin{aligned} H_j &= \begin{cases} Y_i \sim f_Y(y \mid \theta \in \theta_{H_1}),\ Y_i \in T_i \in H_j \\ Y_i \sim f_Y(y \mid \theta \in \theta_{H_0}),\ \text{otherwise} \end{cases} \\ H_\emptyset &= Y_i \sim f_Y(y \mid \theta \in \theta_{H_0}),\ \forall Y_i. \end{aligned} \tag{3.37}$$

If there are a total of J mutually exclusive global hypotheses H_j and the null hypothesis H_ϕ that partition sample space of the cluster image observations, i.e.

$$\sum_{j=1}^{J} \Pr(H_j \mid \mathbf{D_k}) + \Pr(H_\phi \mid \mathbf{D_k}) = 1 \tag{3.38}$$

then the probability that H_j is valid, conditioned on the image observations in the k^{th} cluster is given by:

$$\Pr(H_j \mid \mathbf{D_k}) = \frac{\Pr(\mathbf{D_k} \mid H_j)\Pr(H_j)}{\Pr(\mathbf{D_k})} \tag{3.39}$$

As in (3.21) and (3.31), the above expression can be written as

$$\begin{aligned} \Pr(H_j \mid \mathbf{D_k}) &= \frac{\prod_{T_i \in H_j} \dfrac{\Pr(\mathbf{Y}_{T_i} \mid T_i)\Pr(T_i)}{\Pr(\mathbf{Y}_{T_i} \mid T_\emptyset)\Pr(T_\emptyset)}}{1 + \sum_{H_k} \prod_{T_i \in H_k} \dfrac{\Pr(\mathbf{Y}_{T_i} \mid T_i)\Pr(T_i)}{\Pr(\mathbf{Y}_{T_i} \mid T_\emptyset)\Pr(T_\emptyset)}} \\ &= \frac{\prod_{T_i \in H_j} \exp(L_{T_i} + C_{T_i})}{1 + \sum_{H_k} \prod_{T_i \in H_k} \exp(L_{T_i} + C_{T_i})} \\ &= \frac{\exp(L_{H_j})}{1 + \sum_{H_k} \exp(L_{H_k})} \end{aligned} \tag{3.40}$$

where each hypothesis H_j consists of a set of independent track hypotheses T_i. It follows that

$$\begin{aligned} \Pr(H_j|\mathbf{D_k}) &= \tfrac{\exp(L_{H_j})}{1+\sum_{H_k}\exp(L_{H_k})}, \text{ and} \\ \Pr(H_\emptyset|\mathbf{D_k}) &= \tfrac{1}{1+\sum_{H_k}\exp(L_{H_k})}. \end{aligned} \tag{3.41}$$

3.2.5 Hypothesis Generation

The number of hypotheses generated for a single target can be easily managed. However, when the scope of the problem is expanded to include multiple targets, the number of hypotheses to be managed and evaluated grows rapidly. Thus, an essential component of any multiple hypothesis tracking algorithm is an efficient technique for managing multiple track and global hypotheses. In the following, the generation of track and global hypotheses for the MMHTT algorithm will be outlined and then illustrated with a simple example.

The MMHTT algorithm can be viewed as a three stage processing hierarchy. At the lowest level, multiple multistage hypothesis tests are used to evaluate candidate trajectory segments. When a multistage hypothesis test terminates with an acceptance of the target present hypothesis H_1, the candidate trajectory segment is confirmed and a search for an extension of the confirmed trajectory is initiated in its nominal direction. This stage of the processing hierarchy, designed to meet the signal detection requirements of the MTT system (recall figure 1.2), operates on the incoming preprocessed image data at the image sensor frame rate and outputs short confirmed target trajectory segments.

Algorithm 3 (Constructing Track Hypotheses)

```
Given a confirmed candidate trajectory $t_k$ with observations $\mathbf{Y}_{t_k} = \{y_{k,1}, \ldots, y_{k,n}\}$,
Let the candidate trajectory set origin be $(x_0, y_0)_{t_k}$,
For each target track $T$ whose most recent observation update was $n$ frames ago,
    Let $(x_t, y_t)_T$ be the most recent target position of target track $T$.
    If $(x_0, y_0)_{t_k} = (x_t, y_t)_T$ and $direction(t_k) \subseteq direction(T)$,
        Create a new track hypothesis $T^\star$ with observations $\mathbf{Y}_{T^\star} = \{\mathbf{Y}_T, \mathbf{Y}_{t_k}\}$.
        Evaluate track hypothesis clusters.
Create a new track hypothesis $\hat{T}$ with observations $\mathbf{Y}_{\hat{T}} = \mathbf{Y}_{t_k}$.
Evaluate track hypothesis clusters.
```

In the intermediate processing stage, every trajectory segment detected by the MSHT algorithm is associated with one of the following three hypotheses

1. the trajectory segment is a false alarm,
2. the trajectory segment is an extension of track hypothesis T, or
3. the trajectory segment is a new target detection.

A new track hypothesis is then generated for each track hypothesis for which the candidate trajectory segment is a valid extension. A new track hypothesis is also generated to represent the possibility that the current trajectory segment is a new target observation. This stage of the processing hierarchy generates the raw target tracks required for subsequent state estimation. Note that the data rate at this stage is substantially less than that of its predecessor.

At this point the integrity of each track hypothesis cluster is evaluated. If track hypotheses in different clusters interact, then the clusters are merged.

Algorithm 4 (Evaluate track hypothesis clusters: Cluster Merge)

```
Let $T_N$ be the new track hypothesis.
Set flag.
For each track hypothesis cluster $C_i$,
  For each track hypothesis $T \in C_i$,
    If $\mathbf{Y}_{T_N} \bigcap \mathbf{Y}_T \neq \emptyset$,
      If flag is set,
        Set $N = i$.
        Clear flag.
        Add track hypothesis $T_N$ to cluster $C_i$.
      Else,
        Merge clusters $C_i$ and $C_N$.
If flag is set,
  Create a new track hypothesis cluster $C_{i+1}$.
  Add track hypothesis $T_N$ to cluster $C_{i+1}$.
```

Finally, at the highest level, global hypotheses are generated for each cluster. A global hypothesis is generated for each valid combination of compatible track hypotheses where each global hypothesis represents a different hypothesis concerning the origin of the detected target observations in the track hypothesis cluster. This stage of the processing hierarchy serves a supervisory role evaluating the track hypotheses generated by the intermedidate stage with respect to global tracking constraints. As will be discussed in section 3.3, this stage also fulfills a role in the reduction of computation.

Algorithm 5 (Global hypothesis generation)

```
For each $T \in C$,
  For each global hypothesis $H$ in cluster $C$,
    Set flag.
    For each track hypothesis $T_H \in H$, while flag is set,
      If $\mathbf{Y}_T \bigcap \mathbf{Y}_{T_H} \neq \emptyset$, clear flag.
    If flag is set,
      Create new global hypothesis with track hypotheses $\{T_i \mid T_i \in H\}, T$.
  Create new global hypothesis with track hypothesis $T$.
```

In the following, the preceding procedures for generating track and global hypotheses will be illustrated with a simple example for two targets with intersecting trajectories.

An Example

Consider the tracking scenario illustrated in figure 3.4. The two targets are following intersecting, linear, constant velocity trajectories. To limit the number of hypotheses in this example, only the first new trajectory search, initiated for each target, will be considered. Note that the frame by frame positions of each target are listed in Table 3.1.

The results of a hypothetical 4-stage MSHT designed to detect the two targets are summarized in Table 3.2. Note that Target #1 is initially detected in Frame 1, and Target #2 is initially detected in Frame 2. In both cases, an omnidirectional search of the candidate target trajectories in the candidate trajectory set is inititated at the pixel in which the target was first detected.

In Frame 4, the candidate trajectory from (0,0) in Frame 1 to (3,3) in Frame 4 is confirmed and a search for an extension of this trajectory is initiated in a nominally northeast direction. Since this is the first candidate trajectory to be confirmed, it is not necessary to evaluate this candidate trajectory as a potential update for an existing track hypothesis. Instead, a new track hypothesis cluster $C1$ is created and the candidate trajectory is stored as track hypothesis $T1$ in cluster $C1$.

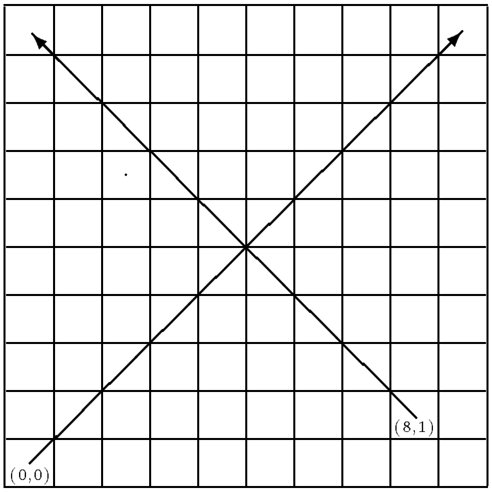

Figure 3.4: Example Target Trajectories

Target #1		Target #2	
Frame	Pixel	Frame	Pixel
1	(0,0)		
2	(1,1)	2	(8,1)
3	(2,2)	3	(7,2)
4	(3,3)	4	(6,3)
5	(4,4)	5	(5,4)
6	(5,5)	6	(4,5)
7	(6,6)	7	(3,6)
8	(7,7)	8	(2,7)
9	(8,8)	9	(1,8)
10	(9,9)	10	(0,9)

Table 3.1: Frame by Frame Target Positions for Tracking Example

Results of Multiple Multistage Hypothesis Tests

Frame	Action	Pixel	Nominal Direction	Target
1	New Test Initiated	(0,0)	○	1
2	New Test Initiated	(8,1)	○	2
4	Target Confirmed	(3,3)		1
	Test Re-initiated	(3,3)	↗	1
	Target Confirmed	(6,3)		2
	Test Re-initiated	(6,3)	↖	2
6	Target Confirmed	(4,5)		2
	Test Re-initiated	(4,5)	↖	2
	Target Confirmed	(5,5)		1
	Test Re-initiated	(5,5)	↗	1
	Target Confirmed	(5,5)		2 (FT)
	Test Re-initiated	(5,5)	↖	2
7	Target Confirmed	(4,6)		2 (FT)
	Test Re-initiated	(4,6)	↖	2 (FT)
	Target Confirmed	(5,6)		1 (FT)
	Test Re-initiated	(5,6)	↗	1 (FT)
8	Target Confirmed	(7,7)		1
	Test Re-Initiated	(7,7)	↗	1
10	Target Confirmed	(0,9)		2
	Target Confirmed	(9,9)		1

Legend: ○ - 360° search, FT - false target

Table 3.2: Detected Trajectory Segments for Tracking Example 1

The candidate trajectory from (8,1) in Frame 2 to (6,3) in Frame 4 is also confirmed in Frame 4, and a search for an extension of this trajectory is initiated in a nominally northwest direction. Recall that the data structure for each candidate trajectory contains the node of the candidate trajectory tree in which the trajectory was confirmed. This node is an index to the candidate trajectory tree data structure which contains the relative offset between the initiation of the multistage hypothesis test and the location of the node in which it was confirmed. Thus, it can be determined that the current candidate trajectory was initiated at (8,1) in Frame 2.

Since there are no current track hypotheses whose most recent observations are from Frame 2, a single track hypothesis is constructed to represent this candidate trajectory. This track hypothesis is then compared to the only existing track hypothesis. The new track hypothesis and track hypothesis $T1$ do not have any image observations in common and hence, do not interact. Therefore, a new track hypothesis cluster is created and the new track hypothesis is stored as track hypothesis $T1$ in $C2$.

Thus after processing Frame 4, there are two track hypothesis clusters, $C1$ and $C2$, with one track hypothesis each. Consequently, each cluster has only one global hypothesis consisting solely of the single track hypothesis. These two clusters and their hypotheses accurately describe the two target trajectories from Frame 1 to Frame 4 (see Table 3.3).

Frame 4						
	Target Track				Track	Global
Cluster	1	2	3	4	Hypothesis	Hypothesis
1	(0,0)	(1,1)	(2,2)	(3,3)	T1	H1
2		(8,1)	(7,2)	(6,3)	T1	H1

Table 3.3: Track Hypotheses after Frame 4

In Frame 6, the candidate trajectory from (5,4) in Frame 5 to (4,5) in Frame 6 is confirmed as a target trajectory segment. Since this candidate trajectory was initiated at (6,3) in Frame 4, both existing track hypotheses, having received their last observation update in Frame 4, are candidates to be updated with the current trajectory. However, only track hypothesis $T1$ in cluster $C2$ terminates at (6,3) in Frame 4, the origin of the current candidate trajectory, and has a direction compatible with the direction of the current candidate trajectory. Thus, track hypothesis $T1$ in cluster $C1$ is removed from consideration and a new track hypothesis is created by concatenating the observations of track hypothesis $T1$ in cluster $C2$ with the observations of the current candidate trajectory.

This new track hypothesis is then compared with the track hypotheses in the existing track hypothesis clusters. The new track hypothesis does not interact with track hypothesis $T1$ in $C1$, but it does interact with track hypothesis $T1$ in $C2$. Therefore, the new track hypothesis is stored as track hypothesis $T2$ in $C2$.

A new track hypothesis is also created to represent the hypothesis that the current candidate trajectory is an observation of a new target. This track hypothesis only contains the observations of the current candidate trajectory and is thus a subset of $T2$ in $C2$. The new track hypothesis interacts with both $T1$ and $T2$ in $C2$ but does not interact with $T1$ in $C1$ and is thus stored as track hypothesis $T3$ in $C2$.

In a similar fashion, the candidate trajectory from (4,4) in Frame 5 to (5,5) in Frame 6 is confirmed as a target trajectory segment originating at (3,3) in Frame 4. This candidate trajectory yields track hypotheses $T2$ and $T3$ in cluster $C1$ (see Table 3.4). At this point both target trajectories have been updated to Frame 6 with the generation of appropriate track hypotheses.

However, a candidate trajectory from (5,4) in Frame 5 to (5,5) in Frame 6 is also confirmed as a target trajectory segment in Frame 6. This trajectory segment originates at (6,3) in Frame 4 and consists of a single observation of Target #2 at (5,4) in Frame 5 and a single observation of Target #1 at (5,5) in Frame 6. Although this is a false target trajectory it consists of a sequence of target observations and hence is accepted as a true target trajectory at the track level.

First, the candidate trajectory is considered as an extension of the two track hypotheses which terminate in Frame 4. Only track hypothesis $T2$ in $C2$ terminates at (6,3) in Frame 4 and has a compatible target direction. Thus, only one track hypothesis is generated by extending track hypothesis $T2$ in $C2$ with the false target trajectory segment.

Frame 6

	Target Track						Track	Global
Cluster	1	2	3	4	5	6	Hypothesis	Hypotheses
1	(0,0)	(1,1)	(2,2)	(3,3)			T1	H1, H5
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	T2	H2
				(3,3)	(4,4)	(5,5)	T3	H3
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)	T4	H4, H5
2		(8,1)	(7,2)	(6,3)			T1	H1
		(8,1)	(7,2)	(6,3)	(5,4)	(4,5)	T2	H2
				(6,3)	(5,4)	(4,5)	T3	H3

Table 3.4: Tracking Hypotheses before Cluster Merge in Frame 6.

This new track hypothesis is then compared with the track hypotheses in the existing clusters. First, the new track hypothesis is compared with the track hypotheses in $C1$. The new track hypothesis shares the observation of Target #1 in Frame 6 with track hypotheses $T2$ and $T3$ in $C1$. Therefore, the new track hypothesis is stored as track hypothesis $T4$ in $C1$.

The new track hypothesis is then compared with the track hypotheses in $C2$. Note that the new track hypothesis shares the observation of Target #2 in Frame 5 with track hypotheses $T2$ and $T3$ in $C2$. Thus, clusters $C1$ and $C2$ are merged. The state of the target tracking hypotheses in Frame 6 before the clusters are merged is summarized in Table 3.4.

Merging the track hypotheses is a simple matter of adding the track hypotheses from $C2$ to $C1$ and renumbering the track hypotheses. In practice, the track hypotheses in each cluster are stored in linked lists (see section 3.4) and the merging of two clusters is a simple matter of changing two data structure pointers. The track hypotheses after the clusters are merged are listed in Table 3.5.

Frame 6							
Cluster	Target Track						Track Hypothesis
	1	2	3	4	5	6	
1	(0,0)	(1,1)	(2,2)	(3,3)			T1
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	T2
				(3,3)	(4,4)	(5,5)	T3
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)	T4
		(8,1)	(7,2)	(6,3)			T5
		(8,1)	(7,2)	(6,3)	(5,4)	(4,5)	T6
				(6,3)	(5,4)	(4,5)	T7
$*$				(6,3)	(5,4)	(5,5)	T8

Table 3.5: Track Hypotheses after Cluster Merge in Frame 6 ($*$ and after Frame 6).

Returning to consideration of the current confirmed candidate trajectory, a new track hypothesis is generated to represent the hypothesis that the current trajectory is an observation of a new target. This track hypothesis consists solely of the image observations from (6,3) in Frame 4 to (5,5) in Frame 6. Since this track hypothesis shares observations of Target #2 with track hypotheses $T4, T5, T6$ and $T7$; and an observation of Target #1 with track hypotheses $T2, T3$ and $T4$, it is only compatible with track hypothesis $T1$. Hence, this track hypothesis is added to the new merged cluster as track hypothesis $T8$.

A total of three candidate target trajectories are accepted in Frame 6. Two of these are valid extensions of the detected trajectories for Target #1 and Target #2, but the third trajectory is a false trajectory whose final observation is *stolen* from Target #1. The observations along this trajectory satisfy the target present hypothesis of the multistage hypothesis test, but lead to the acceptance of a False Track (FT) hypothesis. This track is proposed as a candidate trajectory extension for $T1$ in C2, generating track hypothesis $T4$ as a hypothesized extension of $T5$ and track hypothesis $T8$ as a newly detected target (see Table 3.5). As discussed previously, each detected trajectory segment can be classified as either a false alarm, a new target or a new observation of a current target. For example, classifying the detected trajectory segment $\{(3,3),(4,4),(5,5)\}$

1. as a false alarm suggests the termination of track hypothesis T1,

2. as a continuation of track hypothesis T1 generates hypothesis T2,

3. and as a new target generates track hypothesis T3.

After processing Frame 6, the single track hypothesis cluster contains 8 track hypotheses. If desired, global hypotheses for the observations in this cluster could be generated and evaluated. The resulting set of global hypotheses is enumerated in Table 3.6.

In Frame 7, two candidate trajectory segments are confirmed. Both segments are the result of interactions between Target #1 and Target #2 that generate false track hypotheses. As an illustration of the rapid growth in the number of hypotheses to be evaluated and maintained, the track hypotheses after Frame 7 are summarized in Table 3.7 and the corresponding global hypotheses are enumerated in Table 3.8.

Frame 6	
Global Hypothesis	Track Hypotheses
H1	T1
H2	T2
H3	T3
H4	T4
H5	T1, T4
H6	T5
H7	T6
H8	T7
H9	T1, T5
H10	T1, T6
H11	T1, T7
H12	T2, T5
H13	T2, T6
H14	T2, T7
H15	T3, T5
H16	T3, T6
H17	T3, T7
H18	T8
H19	T1, T8

Table 3.6: Global Track Hypotheses after Frame 6

Frame 7								
	Target Track							Track
Cluster	1	2	3	4	5	6	7	Hypothesis
1	(0,0)	(1,1)	(2,2)	(3,3)				T1
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)		T2
				(3,3)	(4,4)	(5,5)		T3
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)		T4
		(8,1)	(7,2)	(6,3)				T5
		(8,1)	(7,2)	(6,3)	(5,4)	(4,5)		T6
				(6,3)	(5,4)	(4,5)		T7
				(6,3)	(5,4)	(5,5)		T8
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)	(4,6)	T9
				(6,3)	(5,4)	(5,5)	(4,6)	T10
						(5,5)	(4,6)	T11
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(4,5)	(5,6)	T12
				(3,3)	(4,4)	(4,5)	(5,6)	T13

Table 3.7: Track Hypotheses after Frame 7

Frame 7			
Global Hypothesis	Track Hypotheses	Global Hypothesis	Track Hypotheses
H1	T1	H25	T1,T11
H2	T2	H26	T5,T11
H3	T3	H27	T6, T11
H4	T4	H28	T7,T11
H5	T1, T4	H29	T1,T5,T11
H6	T5	H30	T1,T6,T11
H7	T6	H31	T1,T7,T11
H8	T7	H32	T12
H9	T1,T5	H33	T4,T12
H10	T1,T6	H34	T5,T12
H11	T1,T7	H35	T8,T12
H12	T2,T5	H36	T9,T12
H13	T2,T6	H37	T10,T12
H14	T2,T7	H38	T11, T12
H15	T3,T5	H39	T5,T11,T12
H16	T3,T6	H40	T13
H17	T3,T7	H41	T4, T13
H18	T8	H42	T5,T13
H19	T1,T8	H43	T8,T13
H20	T9	H44	T9,T13
H21	T1,T9	H45	T10,T13
H22	T10	H46	T11,T13
H23	T1,T10	H47	T5,T11,T13
H24	T11		

Table 3.8: Global Track Hypotheses after Frame 7

Summary

The preceding example has illustrated the process of track and global hypothesis generation for a simple multitarget tracking scenario. Even for the simple case presented, the total number of hypotheses increases rapidly. In general, it is computationally infeasible to maintain all possible hypotheses. In fact, the number of hypotheses can grow exponentially [7]. Thus, in practice, a number of most probable hypotheses are maintained.

In general, the number of multiple hypotheses maintained is dependent on the computational resources of the MTT system available for target tracking. A number of strategies can be employed to reduce the computational requirements of the MHT approach by selectively pruning and combining hypotheses. By selectively maintaining the most probable hypotheses, a MTT system can balance the need for computational efficiency with the improved error performance provided by maintaining and evaluating additional hypotheses. Strategies for reducing the computational cost of the MHT approach while minimizing the resulting performance loss are discussed in the following section.

3.3 Computational Refinements

In the following, various strategies for reducing the computational cost of the MMHTT algorithm will be discussed. The section begins with the development of a new test initiation procedure for the MMHTT algorithm which exploits knowledge of the current track hypotheses to suppress new MSHTs for targets currently being tracked. Strategies for hypothesis pruning and combining, commonly applied to MHT algorithms, are then discussed and applied to the simple tracking example introduced in the previous section. Acknowledging that it is computationally impractical to maintain and evaluate all possible data association hypotheses, these strategies attempt to minimize the performance loss associated with an evaluation of an incomplete set of hypotheses by selectively generating and maintaining a set of most probable hypotheses.

3.3.1 Track Initiation

An effective means for reducing the computational cost of the MMHTT algorithm is to reduce the number of multistage hypothesis tests initiated to evaluate candidate trajectory segments in each image frame. Reducing the number of new candidate trajectory searches leads to a reduction in the number of confirmed target tracks and consequently, the number of track and global hypotheses to be generated and evaluated. The challenge is to selectively reduce the number of new target trajectory searches in a manner which minimizes the resulting loss in detection and tracking performance.

In a typical MTT system, track initiation is the most resource intensive process. The goal of the track initiation stage is to reduce the demand for computational resources without compromising the overall system detection performance. Typically, the track initiation functions are served by a dedicated signal processor which can meet the high data rate of the incoming image signal, producing a relatively small set of potential targets/image frame for evaluation by a higher level, general purpose track hypothesis processor. The signal processor must detect enough true targets to satisfy the system's probability of detection requirements and yet reject sufficient false targets to avoid saturating the post-processor and to limit the overall system false alarm rate to an acceptable level.

Three strategies are proposed to accomplish this task:

1. Evaluation of current candidate trajectory hypotheses.

2. Rate-constrained target detection.

3. Employment of an auxiliary region of interest (ROI) process.

Evaluation of Current Candidate Trajectory Hypotheses

In the original implementation of the MSHT algorithm, the log-likelihood ratio function $L(y)$ was evaluated for the observed image intensity y of every pixel in every image frame [40]. If the test statistic $L(y)$ for a given pixel exceeded the upper test threshold $\hat{a}$, then a target was immediately declared present. If the test statistic $L(y)$ failed to exceed the lower test threshold $\hat{b}$, then the pixel was removed from further consideration. However, if the test statistic was bounded by the first stage MSHT test thresholds, then an omnidirectional evaluation of the trajectories in the candidate trajectory set for that pixel was initiated.

Although this approach ensures that all candidate trajectories are evaluated, it leads to an evaluation of a significant number of redundant candidate trajectories. In particular, for a typical MSHT, with a high probability of detection, this procedure results in multiple detected target trajectories for a single target track, many of which differ by a single observation. Although in the original MSHT implementation, these redundant trajectories provided a rough tracking capability, through repeated target detection, the introduction of the reinitiation of detected target trajectories in the MMHTT algorithm restricts the need for track initiation to the detection of new targets.

It is worth noting that for typical MSHT test designs, the expected test length and hence the computational cost of MSHTs evaluating candidate trajectories with target observations is significantly greater than that of MSHTs evaluating candidate trajectories without target observations (see Chapter 4). In addition, the probability of accepting a false candidate target trajectory is significantly increased if its first observation contains a target (see Chapter 4). Thus, by suppressing the initiation of new candidate trajectory searches for pixels currently hypothesized as members of a target trajectory, the computational cost of the track initiation stage can be reduced with an accompanying reduction in the number of false alarms.

Consequently, the initiation of new trajectory searches in the MMHTT algorithm is inhibited at image pixels which coincide with the location of current undecided trajectories. New trajectories are only initiated when there is insufficient evidence to support the hypothesis that the current pixel, $I[\mathbf{x}, t]$, is an observation of an old target in the current undecided trajectory list. The new target/old target decision is formulated as a simple Bayesian hypothesis testing problem conditioned on the observations of all the candidate trajectories in the pixel's undecided trajectory list.

Defining the set of candidate trajectories in the current pixel's undecided trajectory list as

$$\Omega = \{t_k \mid y_{k,t} = I[\mathbf{x}, t]\} \tag{3.42}$$

there are two possible hypotheses regarding the observation $I[\mathbf{x}, t]$:

$$\begin{aligned} H_{NT}: &\quad I[\mathbf{x}, t] \text{ is an observation from a new target} \\ H_{OT}: &\quad I[\mathbf{x}, t] \text{ is an observation from an old target} \end{aligned} \tag{3.43}$$

The *new target* hypothesis, H_{NT}, implies that all the observations along the trajectories in the current pixel's undecided trajectory list are distributed as

$$Y \sim f_Y(y \mid \theta \in \theta_{H_0}),\ \forall y \in t_i \in \Omega \tag{3.44}$$

while the *old target* hypothesis, H_{OT}, implies that the current observation, $I[\mathbf{x}, t]$, is an observation of a target which is also present in one of the candidate trajectories in the current undecided trajectory list.

$$\Pr(H_{OT}) = \Pr(\bigcap_j H_{OT_j}) \tag{3.45}$$

where

$$H_{OT_j} : Y \sim \begin{cases} f_Y(y \mid \theta \in \theta_{H_1}),\ \forall y \in t_j \\ f_Y(y \mid \theta \in \theta_{H_0}),\ \forall y \in t_i \in \Omega\ t_i \neq t_j \end{cases} \tag{3.46}$$

Recall that no more than one target is assumed present in any image observation. Hence, acceptance of each of the candidate trajectories represents a set of mutually exclusive hypotheses.

Defining the image observations along the candidate trajectories in the undecided trajectory list as

$$\mathbf{Y}_\Omega \bigcup_{t_i \in \Omega} \mathbf{Y}_{t_i} \tag{3.47}$$

the Bayes decision rule for (3.43), with a uniform cost assignment [63], accepts H_{NT} if

$$\begin{aligned} \Pr(\mathbf{Y}_\Omega \mid H_{NT}) \Pr(H_{NT}) &\geq \Pr(\mathbf{Y}_\Omega \mid H_{OT}) \Pr(H_{OT}) \\ \Pr(\mathbf{Y}_\Omega \mid H_{NT}) \Pr(H_{NT}) &\geq \textstyle\sum_{t_i} \Pr(\mathbf{Y}_\Omega \mid H_{OT_i}) \Pr(H_{OT_i}) \\ 1 &\geq \textstyle\sum_{t_i} \frac{\Pr(\mathbf{Y}_{t_i} \mid H_1)}{\Pr(\mathbf{Y}_{t_i} \mid H_0)} \frac{\Pr(H_{OT_i})}{\Pr(H_{NT})} \end{aligned} \tag{3.48}$$

and assuming that

$$\Pr(H_{OT_i}) = \Pr(H_{OT_j}),\ \forall t_i, t_j \in \Omega \tag{3.49}$$

the new target decision rule accepts H_{NT} if

$$\sum_{t_i} \frac{\Pr(\mathbf{Y}_{t_i} \mid H_1)}{\Pr(\mathbf{Y}_{t_i} \mid H_0)} \leq \frac{\Pr(H_{NT})}{\Pr(H_{OT_i})} \tag{3.50}$$

which can be expressed as

$$\sum_{t_i} \exp(L_{t_i}) \leq \tau_{NT} \tag{3.51}$$

where τ_{NT} is the ratio of the prior probability that all the observations in $\mathbf{Y}_\Omega$ are target free to the prior probability that one of the candidate trajectories coincides with a true target trajectory. This decision rule minimizes the probability of error and maximizes the *a posteriori* probability that the selected hypothesis is correct given $\mathbf{Y}_\Omega$.

In the absence of domain dependent information regarding the expected target density, which could be used to estimate the prior probabilities $\Pr(H_{NT})$ and $\Pr(H_{OT_i})$, we assume that $\Pr(H_{NT}) = \Pr(H_{OT_i})$ and thus $\tau_{NT} = 1$. This is a maximum likelihood approach. Alternatively, an estimate for τ_{NT} can be made by monitoring the current ratio of new targets to old targets, as determined by the most probable global hypotheses.

Rate-Constrained Detection

In applications where the available computational resources (time, cpu, memory) are constrained, the system detection performance may be compromised when the rate of target detection exceeds the detection system's processing rate. In order to prevent system overload under these conditions, potential targets must be randomly ignored. Recently, it has been shown that for a two-stage detection system, where a simple first stage detector screens potential targets for a computationally-intensive but reliable second stage, the overall system detection probability is maximized if the first stage is designed using a *rate-constraint* criterion which matches the detection rate of the first stage to the processing rate of the second stage [87].

In this context, the track initiation stage of the MMHTT algorithm can be viewed as a two-stage detection system where the lower first stage MSHT threshold $\hat{b}$ implements a simple first stage detector and the complete MSHT algorithm acts as a computationally expensive but reliable second stage detector. The rate-constraint is easily incorporated by replacing the lower first stage MSHT threshold with an adaptive threshold

$$\tau_{RC} = \max(\hat{b}, T_{RC}). \tag{3.52}$$

Thus, new trajectories are only initiated at pixels where the feature detection likelihood ratio, $f(I[\mathbf{x},t])$, exceeds a threshold T_{RC} which is adaptively computed using the rate-constrained detection criterion,

$$\pi\beta(T_{RC}) + (1-\pi)\alpha(T_{RC}) = \rho \tag{3.53}$$

where π is the *a priori* probability that a *new* target is present, β is the detection probability given T_{RC}, α is the probability of false alarm given T_{RC} and ρ is the maximum managable rate of *new* target detections given the current computational load.

Note that T_{RC} can only take a finite number of discrete values due to the discrete nature of the image observations. Consequently, for specified statistical models of the feature and the image background, $\beta(T_{RC})$ and $\alpha(T_{RC})$ can be pre-computed and stored in a lookup-table for real-time processing. Furthermore, the *a priori* probability π can be estimated from the observed detection rate $\hat{\rho}$.

Auxilliary Region of Interest Process

For certain applications, such as the feature correspondence tracking system discussed in Chapter 5, auxiliary information may be used to control the track initiation process. If auxiliary information is available which can reliably partition each image frame into background image regions and regions of interest for target detection and tracking, this information can be used as a global mask to inhibit track initiation in background regions of the current image frame. For example, as will be discussed in Chapter 5, the detection and tracking of feature point correspondences for moving objects in a stationary image environment can employ a change detection process to identify and isolate those regions of each image frame for which there has been no statistically significant deviation from an adaptive reference image described in section 5.3.2. This allows the system to focus its computational resources on detecting and tracking feature points (targets) in those regions of the image most likely to contain moving objects.

3.3.2 Pruning and Combining Hypotheses

The following section outlines several standard MHT techniques for pruning and combining hypotheses [7]. These techniques are commonly used to manage the computational complexity of multiple hypothesis tracking algorithms and provide a convenient means to control the computational cost of the MMHTT algorithm. Acknowledging that computational resource limitations restrict the number of hypotheses that can be effectively managed and evaluated, these techniques attempt to identify those tracking hypotheses which are most probable.

Competing hypotheses are evaluated by comparing their respective score functions (see sections 3.2.3 and 3.2.4). Those hypotheses with the highest scores are retained and those with lower scores are deleted. The probability that each of the retained hypotheses is valid can then be re-evaluated relative to the total probability of the remaining hypotheses. It is commonly recognized that by ignoring a number of tracking hypotheses with a small but nonzero probability the *optimality* of the Bayesian multiple hypothesis approach is compromised [35, 7]. However, the resultant performance loss is typically small, and dependent on the total probability of the deleted hypotheses.

In the following, several strategies for hypothesis reduction will be outlined and then applied to the tracking example presented in section 3.2.5.

Combining Track Hypotheses

Consider, for a moment, two track hypotheses which have their last N observations in common. As time progresses, the distinct origin of each track hypothesis becomes less and less relevant to the decision processes of the MTT system. At some point, the two hypotheses are essentially redundant with respect to current tracking decisions and the two hypotheses can be combined.

This concept is formalized in what is commonly known as the *N-scan rule* [7]. Hypotheses which represent the same trajectory, within the resolution requirements of a given application, are combined by combining hypotheses whose last N observations are the same. The value of N is application dependent and is a function of the desired accuracy and temporal history of target observations.

Pruning Track Hypotheses

As discussed in section 3.2.4, each global tracking hypothesis consists of a set of compatible track hypotheses, where in general, each track hypothesis may be a member of more than one global hypothesis. Following Blackman et al. [35], we define the *global* probability that a given track hypothesis T in the k^{th} cluster C_k is valid as:

$$\Pr(T \mid D_k) = \sum_{j=1}^{J} \Pr(T \mid H_j, D_k) \Pr(H_j \mid D_k) \tag{3.54}$$

where

$$\Pr(T \mid H_j, D_k) = \begin{cases} 1, & \text{if } T \in H_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.55}$$

Note that this is just the sum of the probabilities that the global hypotheses, of which T is a member, are valid.

This global evaluation of the probability that the track hypothesis T is valid, is more accurate than the local evaluation described in section 3.2.3, as it incorporates all the data association hypotheses regarding its constituent image observations. Thus, an evaluation of the global track hypothesis probabilities can identify track hypotheses which appear highly probable in a local analysis but which are members of improbable global hypotheses. These track hypotheses are candidates for hypothesis pruning.

There are two basic approaches to pruning track hypotheses [7]. One technique is to rank all current track hypotheses based on an evaluation of their global probabilities and maintain the M most probable hypotheses. This technique guarantees a constant computational load for hypothesis maintenance. Alternatively, one can prune unlikely global hypotheses and by association prune their constituent track hypotheses. For example, Blackman has proposed a pruning procedure in which the (N+1)'th hypothesis in a ranked list of global hypotheses is deleted if it satisfies

$$\sum_{j=1}^{N} L_{H_j} - N L_{H_{N+1}} > T_S \tag{3.56}$$

where N is the total number of alternate hypotheses, and T_S is a decision threshold which may adapt to the current computational load of the host system [7]. This procedure prunes track hypotheses which are not members of retained global hypotheses.

Splitting Clusters

After the track hypotheses have been pruned, the track hypothesis clusters should be re-evaluated. The removal of one or more track hypotheses from a given cluster may allow the cluster to be split into two or more sets of non-interacting tracks. If this is the case, then the cluster is split into smaller clusters accordingly.

Algorithm 6 (Track hypothesis cluster evaluation: Cluster Split)

```
Set flag.
For each Cluster $C$, from which a track hypothesis was pruned,
  Let $NT$ be the number of track hypotheses in Cluster $C$.
  For track hypothesis $T_j \in C$ $j$ := 1 to $NT$,
    For each new cluster $\hat{C}_k$,
      For each track hypothesis $T \in \hat{C}_k$,
        If $\mathbf{Y}_{T_j} \bigcap \mathbf{Y}_T \neq \emptyset$,
          If flag is set,
            Move $T_j$ from $C_i$ to $\hat{C}_k$.
            Set $temp = k$ and clear flag.
          Else merge $\hat{C}_{temp}$ and $\hat{C}_k$.
    If flag is set,
      Create new cluster $\hat{C}$.
      Move $T_j$ from $C$ to $\hat{C}$.
      For each track hypothesis $T_k \in C$, $k > j$,
        For each track hypothesis $T \in \hat{C}$,
          If $\mathbf{Y}_{T_k} \bigcap \mathbf{Y}_T \neq \emptyset$,
            Move $T_k$ from $C$ to $\hat{C}$.
            For each new cluster $\hat{C}_k$,
              For each track hypothesis $T \in \hat{C}_k$,
                If $\mathbf{Y}_{T_k} \bigcap \mathbf{Y}_T \neq \emptyset$,
                  Merge $\hat{C}$ and $\hat{C}_k$.
```

An Example

Returning to the example tracking scenario presented in section 3.2.5, the effects of hypothesis pruning will be examined by reworking the generation and evaluation of the tracking hypotheses summarized in Tables 3.3 to 3.8. For the purposes of this example, it will be assumed that the alternative hypothesis pruning procedure is applied with $T_S = 4$.

Reconsider the track hypotheses after Frame 6, as summarized in Table 3.5. As discussed in section 3.2.3, each track hypothesis has a track hypothesis score L_T which is the sum of the scores of its constituent candidate trajectory segments. The track hypothesis scores for the eight track hypotheses in existence after Frame 6 are listed in Table 3.9.

Frame 6	
Track Hypothesis	Score
T1	3.16
T2	6.52
T3	3.36
T4	8.02
T5	3.70
T6	6.92
T7	3.22
T8	4.32

Table 3.9: Track Hypothesis Scores after Frame 6

Given the scores for each individual track hypothesis, the scores for the global hypotheses enumerated in Table 3.6 can also be computed. Assuming equal priors for the track hypotheses, ($C_T = 0 \; \forall T$), the global hypothesis scores are simply the sums of the scores of their constituent track hypotheses (see Table 3.10).

After applying the pruning procedure proposed by Blackman [7], with $T_S = 4$, only the top two global hypotheses $H13$ and $H5$ are retained. Hypothesis $H13$, consisting of track hypotheses $T2$ and $T6$, correctly identifies the two target trajectories and

Frame 6					
Global	Global		Global	Global	
Hypothesis	Score	$\Pr(H_i \mid \mathbf{D_k})$	Hypothesis	Score	$\Pr(H_i \mid \mathbf{D_k})$
H1	3.16	3×10^{-5}	H11	6.38	7×10^{-4}
H2	6.52	8×10^{-4}	H12	10.22	0.0317
H3	3.36	3×10^{-5}	H13	13.44	0.793
H4	8.02	3.5×10^{-5}	H14	9.74	0.0196
H5	11.18	0.0828	H15	7.06	1×10^{-3}
H6	3.70	5×10^{-5}	H16	10.28	0.0336
H7	6.92	1×10^{-3}	H17	6.58	8×10^{-4}
H8	3.22	3×10^{-5}	H18	4.32	9×10^{-5}
H9	6.86	1×10^{-3}	H19	7.48	2×10^{-3}
H10	10.08	0.0276			

Table 3.10: Global Hypothesis Scores after Frame 6

is ranked as the most likely hypothesis with $\Pr(H_{13} \mid \mathbf{D_1}) = 0.905$ after pruning. Hypothesis $H5$, consisting of track hypotheses $T1$ and $T4$, is retained as an alternate global hypothesis with $\Pr(H_5 \mid \mathbf{D_1}) = 0.0945$ after pruning. Since track hypotheses $T1, T2, T4$ and $T6$ are the only track hypotheses which are members of at least one of the retained global hypotheses, the remaining track hypotheses are deleted.

At this point the track hypotheses in the cluster are re-evaluated to determine if it is possible to split the cluster. Note that track hypothesis $T1$ interacts with track hypothesis $T2$, track hypothesis $T2$ interacts with track hypothesis $T4$ and track hypothesis $T4$ interacts with track hypothesis $T6$. Thus, all the track hypotheses in the cluster interact with each other either directly or indirectly and the cluster cannot be split.

Continuing on to Frame 7, the renumbered track hypotheses after Frame 7, with hypothesis pruning, are listed in Table 3.11 and the renumbered global hypotheses after Frame 7 are listed in Table 3.12. In this instance, the pruning procedure retains the top 3 hypotheses $H15, H7$ and $H13$. Note that the most likely hypothesis $H15$ ($\Pr(H_{15} \mid \mathbf{D_k}) = 0.72$ after pruning) consists of the true target trajectory for Target #2 and a false target trajectory for Target #1. A standard single frame data association algorithm would incorrectly accept hypothesis $H15$ and delete the true track hypothesis for Target #1, $T2$. However, the MMHTT algorithm maintains multiple hypotheses including the correct global hypothesis $H7$ ($\Pr(H_7 \mid \mathbf{D_k}) = 0.17$ after pruning) and another alternate hypothesis $H13$ ($\Pr(H_{13} \mid \mathbf{D_k}) = 0.11$ after pruning). Consequently, track hypotheses $T1, T2, T3, T4, T6$ and $T7$ are retained. Note that all of these track hypotheses interact with each other either directly or indirectly and hence, the cluster cannot be split.

In Frame 8, a single candidate trajectory segment is confirmed (see Table 3.2). This segment is the result of a test initiated at (5,5) in Frame 6 and confirmed at (7,7) in Frame 8. The renumbered track hypotheses which survived the pruning procedure at the end of Frame 7 and the track hypotheses created as a result of the current confirmed trajectory segment are listed in Table 3.13 and the resulting global hypotheses are listed in Table 3.14.

The pruning procedure after Frame 8, selects the two most likely hypotheses $H13$

Frame 7									
	Target Track							Track	
Cluster	1	2	3	4	5	6	7	Hypothesis	Score
1	(0,0)	(1,1)	(2,2)	(3,3)				T1	3.16
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)		T2	6.52
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)		T3	8.02
		(8,1)	(7,2)	(6,3)	(5,4)	(4,5)		T4	6.92
		(8,1)	(7,2)	(6,3)	(5,4)	(5,5)	(4,6)	T5	4.32
						(5,5)	(4,6)	T6	2.98
	(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(4,5)	(5,6)	T7	6.88
				(3,3)	(4,4)	(4,5)	(5,6)	T8	3.72

Table 3.11: Track Hypotheses after Frame 7

Frame 7							
Hypotheses				Hypotheses			
Global	Track	Score	$\Pr(H_i \mid \mathbf{D_k})$	Global	Track	Score	$\Pr(H_i \mid \mathbf{D_k})$
H1	T1	3.16	5×10^{-6}	H12	T4,T6	9.90	4×10^{-3}
H2	T2	6.52	1.5×10^{-4}	H13	T1,T4,T6	13.06	0.105
H3	T3	8.02	7×10^{-4}	H14	T7	6.88	2×10^{-4}
H4	T1,T3	11.18	0.0161	H15	T3,T7	14.9	0.663
H5	T4	6.92	2×10^{-4}	H16	T5,T7	11.20	0.0164
H6	T1,T4	10.08	5×10^{-3}	H17	T6,T7	9.86	4×10^{-3}
H7	T2,T4	13.44	0.154	H18	T8	3.72	9×10^{-6}
H8	T5	4.32	2×10^{-5}	H19	T3,T8	11.74	0.0281
H9	T1,T5	7.48	4×10^{-4}	H20	T5,T8	8.04	7×10^{-4}
H10	T6	2.98	4×10^{-6}	H21	T6,T8	6.70	2×10^{-4}
H11	T1,T6	6.14	1×10^{-4}				

Table 3.12: Global Track Hypotheses after Frame 7

Frame 8									
Target Track								Track	
1	2	3	4	5	6	7	8	Hypothesis	Score
(0,0)	(1,1)	(2,2)	(3,3)					T1	3.16
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)			T2	6.52
	(8,1)	(7,2)	(6,3)	(5,4)	(5,5)			T3	8.02
	(8,1)	(7,2)	(6,3)	(5,4)	(4,5)			T4	6.92
					(5,5)	(4,6)		T5	2.98
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(4,5)	(5,6)		T6	6.88
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	(6.6)	(7,7)	T7	9.64
					(5,5)	(6.6)	(7,7)	T8	3.12

Table 3.13: Track Hypotheses after Frame 8

Frame 8							
Hypotheses				Hypotheses			
Global	Track	Score	$\Pr(H_i \mid \mathbf{D_k})$	Global	Track	Score	$\Pr(H_i \mid \mathbf{D_k})$
H1	T1	3.16	1×10^{-6}	H12	T4,T5	9.90	1×10^{-3}
H2	T1,T3	11.18	3.5×10^{-3}	H13	T4,T7	16.56	0.762
H3	T1,T4	10.08	1×10^{-3}	H14	T1,T4,T8	13.20	0.0265
H4	T1,T5	6.14	2×10^{-5}	H15	T4,T8	10.04	1×10^{-3}
H5	T1,T8	6.28	3×10^{-5}	H16	T5	2.98	1×10^{-6}
H6	T2	6.52	3×10^{-5}	H17	T5,T6	9.86	9×10^{-4}
H7	T2,T4	13.44	0.0336	H18	T6	6.88	5×10^{-5}
H8	T3	8.02	1×10^{-4}	H19	T6,T8	10.00	1×10^{-3}
H9	T3,T6	14.90	0.145	H20	T7	9.64	7.5×10^{-4}
H10	T4	6.92	5×10^{-5}	H21	T8	3.12	1×10^{-6}
H11	T1,T4,T5	13.06	0.0230				

Table 3.14: Global Track Hypotheses after Frame 8

($\Pr(H_{13} \mid \mathbf{D_k}) = 0.84$ after pruning) and $H9$ ($\Pr(H_9 \mid \mathbf{D_k}) = 0.16$ after pruning). Note that neither hypothesis correctly reflects both target trajectories but that both true target trajectories are retained. An evaluation of the retained track hypotheses reveals that the cluster cannot yet be split.

In Frame 10, two candidate trajectory segments are confirmed. The first is the result of a multistage hypothesis test initiated at (4,5) in Frame 6 and confirmed at (0,9) in Frame 10. The second is the result of a multistage hypothesis test initiated at (7,7) in Frame 8 and confirmed at (9,9) in Frame 10. The resulting track hypotheses are listed in Table 3.15 and the corresponding global hypotheses are listed in Table 3.16.

Frame 10											
Target Track										Track	
1	2	3	4	5	6	7	8	9	10	Hypothesis	Score
	(8,1)	(7,2)	(6,3)	(5,4)	(5,5)					T1	8.02
	(8,1)	(7,2)	(6,3)	(5,4)	(4,5)					T2	6.92
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(4,5)	(5,6)				T3	6.88
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	(6.6)	(7,7)			T4	9.64
	(8,1)	(7,2)	(6,3)	(5,4)	(4,5)	(3,6)	(2,7)	(1,8)	(0,9)	T5	9.87
					(4,5)	(3,6)	(2,7)	(1,8)	(0,9)	T6	2.95
(0,0)	(1,1)	(2,2)	(3,3)	(4,4)	(5,5)	(6.6)	(7,7)	(8,8)	(9,9)	T7	12.59
							(7,7)	(8,8)	(9,9)	T8	2.95

Table 3.15: Track Hypotheses after Frame 10

After pruning, only hypotheses $H16, H7$ and $H13$ are retained. These global hypotheses are constructed from the four track hypotheses $T2, T4, T5$ and $T7$. Note that track hypotheses $T2$ and $T5$ interact with each other but do not interact with track hypotheses $T4$ and $T7$, and vice versa. Thus, the cluster can now be split into two track hypothesis clusters. One cluster contains track hypotheses $T2$ and $T5$ and the other cluster contains track hypotheses $T4$ and $T7$.

The global hypotheses for each of the new track hypothesis clusters now consist of single track hypotheses as in Frame 4 (see Table 3.17). Note that the algorithm has

Frame 10							
Hypotheses		Score	$\Pr(H_i \mid \mathbf{D_k})$	Hypotheses		Score	$\Pr(H_i \mid \mathbf{D_k})$
Global	Track			Global	Track		
H1	T1	8.02	5×10^{-7}	H13	T4,T5	19.51	0.0468
H2	T1,T3	14.9	5×10^{-4}	H14	T4,T6	12.59	5×10^{-5}
H3	T1,T6	10.97	9×10^{-6}	H15	T5	9.87	3×10^{-6}
H4	T1,T8	10.97	9×10^{-6}	H16	T5,T7	22.46	0.893
H5	T2	6.92	2×10^{-7}	H17	T5,T8	12.82	6×10^{-5}
H6	T2,T4	16.56	2×10^{-3}	H18	T6	2.95	3×10^{-9}
H7	T2,T7	19.51	0.0468	H19	T6,T7	15.54	9×10^{-4}
H8	T2,T8	9.87	3×10^{-6}	H20	T1,T6,T8	13.92	2×10^{-4}
H9	T3	6.88	1.5×10^{-7}	H21	T6,T8	5.90	6×10^{-8}
H10	T1,T3,T8	17.85	9×10^{-3}	H22	T7	12.59	5×10^{-5}
H11	T3,T8	9.83	3×10^{-6}	H23	T8	2.95	3×10^{-9}
H12	T4	9.64	2×10^{-6}				

Table 3.16: Global Track Hypotheses after Frame 10

correctly resolved the two distinct target trajectories with the reception of additional observations of each target. The algorithm's ability to correct the association errors it committed in Frames 7 and 8 is a direct result of its ability to maintain and evaluate alternate tracking hypotheses.

Frame 10				
	Hypotheses			
Cluster	Global	Track	Score	$\Pr(H_i \mid \mathbf{D_k})$
1	H1	T2	6.92	0.05
	H2	T5	9.87	0.95
2	H1	T4	9.64	0.05
	H2	T7	12.59	0.95

Table 3.17: Global Track Hypotheses after Cluster Split

3.4 Implementation Notes

The data processing algorithms used to manage the multiplicity of detection and tracking hypotheses in the MMHTT algorithm can be efficiently implemented using standard dynamic data structures [7, 88]. In particular, the linked list is a convenient construct for changing the logical organization of a data structure without affecting its physical memory storage. Conceptually, a linked list is simply a list of data structures or records which are linked by a construct called a pointer. A pointer is simply a variable which contains a memory address, in this case the memory address of a particular data record. The primary advantage of these constructs is that many of the MMHTT data processing algorithms only affect the logical organization of the track records and can thus be implemented as simple pointer operations without directly manipulating the data records themselves.

Linked lists are used to manage multiple hypotheses at every stage of the MMHTT processing hierarchy. At the lowest level, candidate trajectories are stored in linked lists relative to their hypothesized location in the current frame. These candidate trajectories are used to form track hypotheses which are stored in linked lists of independent track hypotheses or track hypothesis clusters. Meanwhile, at the highest level, each global hypothesis contains a linked list of track hypotheses and is itself contained in a linked list of global hypotheses for a particular track hypothesis cluster. Each track hypothesis cluster thus contains a linked list of member track hypotheses and a linked list of global hypotheses generated from the member track hypotheses. Each cluster is itself stored in a linked list of track hypothesis clusters. In the following, the dynamic data structures used to implement the MMHTT algorithm will be briefly outlined and discussed.

As discussed in section 3.1.2, the candidate trajectory records generated by the lowest level of the MMHTT processing hierarchy are stored in linked lists. Each candidate trajectory is stored in the linked list associated with its hypothesized location in the current frame. Thus, for each pixel location, there is a linked list of candidate trajectories which pass through that pixel in the current frame.

For each new image frame, the candidate trajectories, in the linked list for each pixel, are updated with the new image observation and then evaluated with the next stage of their respective multistage hypothesis tests. If the target absent hypothesis is accepted, the candidate trajectory is removed from the list and its memory returned to the system. If the target present hypothesis is accepted, the candidate trajectory is passed to the tracking stage and a new candidate trajectory search is initiated in the current pixel. However, if the multistage hypothesis test is undecided, the children of the current candidate trajectory are obtained from the candidate trajectory data structure, using the node id in the candidate trajectory record (see sections 3.1.1 and 3.1.2). New candidate trajectory data records are then created and added to the linked list of the appropriate pixel in the subsequent frame. Thus, the detection stage of the MMHTT algorithm operates on the preprocessed image data and the undecided candidate trajectories stored in the linked lists for each image pixel to generate confirmed target trajectory segments for track hypothesis generation and evaluation.

The basic element of the tracking stage of the MMHTT algorithm is the track record. Each track record summarizes the essential details of a hypothesized target within the following fields:

1. a pointer to the previous track record in the track master list,
2. a pointer to the next track record in the track master list,
3. a pointer to the previous track record in the track hypothesis cluster list,
4. a pointer to the next track record in the track hypothesis cluster list,
5. the track hypothesis score, L_{T_k},
6. the track's nominal direction, and
7. a list of the last N target observations for the N-scan combining rule.

In addition, a track record for a complete MTT system may also contain the current target state estimate.

A new track record is created for every track hypothesis that is generated. The records themselves are stored in a master list of track hypotheses and then referenced by track pointers in the track hypothesis cluster and global hypothesis data structures. This allows the data memory for the track records to be managed independently of the logical organization of the track records into track hypothesis clusters and global hypotheses.

As discussed in section 3.2.4, the set of track hypotheses are organized into subsets of non-interacting track hypotheses or track hypothesis clusters. Each track hypothesis cluster is a dynamic data structure consisting of a list of track hypotheses and a list of global hypotheses in a linked list of other track hypothesis clusters. Thus, each cluster data record contains the following fields:

1. a pointer to the previous cluster in the cluster list,
2. a pointer to the next cluster in the cluster list,
3. a linked list of pointers to the track hypothesis records in the track master list,
4. and a linked list of global hypotheses.

Each global hypothesis is simply a logical collection of track hypotheses. Thus, each global hypothesis record simply contains:

1. a pointer to the previous global hypothesis in the cluster,
2. a pointer to the next global hypothesis in the cluster,
3. the global hypothesis score L_H, and
4. a linked list of pointers to the track hypothesis records in the track master list.

As candidate trajectories are confirmed, track records are created for each new track hypothesis (see Algorithm 3) and added to the track master list. With the addition of each new track hypothesis, the track hypothesis clusters are re-evaluated (see Algorithm 4). If the new track hypothesis interacts with one or more track hypotheses from a given track hypothesis cluster, then a pointer to the new track hypothesis track record is added to the cluster track list. If the new track hypothesis interacts with track hypotheses from more than one track hypothesis cluster then the clusters are merged. Merging two clusters is a simple matter of adding the track list of one cluster to the track list of the other cluster and removing the first cluster from the cluster list. Finally, if the new track hypothesis does not interact with any of the track hypotheses in the current track hypothesis clusters, then a new track hypothesis cluster record is created and added to the cluster list. A pointer to the new track hypothesis record is then added to the track list of the newly created track hypothesis cluster record.

Global hypotheses can be generated for each cluster as desired for a global evaluation of the track hypotheses or for hypothesis pruning (see Algorithm 5). If after hypothesis pruning the track hypotheses in the track hypothesis cluster can be split into two or more smaller track hypothesis clusters (see Algorithm 6), the cluster is removed from the cluster list and new cluster records are created for each of the new smaller clusters. These clusters are then added to the cluster list and processing resumes.

3.5 Summary

A new algorithm, Multiple Multistage Hypothesis Test Tracking, for detecting and tracking point-source targets in a sequence of digital images has been developed. The MMHTT algorithm transforms a sequence of preprocessed digital images into a set of target tracks suitable for state/motion estimation. At the lowest level, the algorithm behaves as a spatial array of independent detectors with local communications. Each detector implements a robust and efficient sequential detection algorithm (truncated SPRT) by updating the multistage hypothesis tests for the local candidate target trajectories. Once confirmed, these target trajectory segments are combined to form target track hypotheses to describe the observed dynamics of the detected targets over an extended number of frames.

Each hypothesized target trajectory is approximated by a sequence of short, linear, constant velocity trajectory segments, generated by successive candiate trajectory searches. This provides the algorithm with an efficient means of tracking maneouvring targets for an extended number of image frames with a minimal assumption of local trajectory linearity. Finally, at a global level, known physical constraints can be applied to generate and evaluate globally consistent sets of track hypotheses and to resolve data association decisions which are ambiguous at a local level.

Algorithm 7 (Multiple Multistage Hypothesis Test Tracking)

```
Construct hierarchical lookup-table for candidate trajectory set.
Construct hypothesis test threshold lookup-table indexed by test stage.
Initialize undecided candidate trajectory lists for every image pixel location.
For each preprocessed image frame,
    Evaluate the new image observation at each pixel for test initiation (see section 3.3.1).
    Update the multistage hypothesis tests for each candidate trajectory (see Algorithm 5).
    Update track hypotheses for each cluster (see Algorithms 3 and 4).
    Generate global hypotheses (see Algorithm 5).
    Prune and combine track hypotheses (see section 3.3.2).
```

Chapter 4

Performance Analysis

The following chapter develops procedures for evaluating the performance of the MMHTT algorithm. Performance expressions are developed from models of the background clutter and detector noise, and used to develop simple computational procedures to predict the algorithm's performance for arbitrary multitarget scenarios. In general, a thorough performance analysis requires extensive computer simulations under realistic operating conditions. In fact, the development of definitive performance measures for multitarget tracking is still an open problem [89, 90]. Consequently, the following discussion is limited to the development of performance analysis procedures that are a direct function of the MMHTT algorithm.

The keystone in the development of the MMHTT algorithm is the application of the truncated SPRT to the binary hypothesis testing problem of (2.14), with independent observations. An analysis of this problem yields useful measures of system performance. In particular, an analysis of the performance of a truncated SPRT, as a decision rule for (2.14), yields three statistics

1. the probability of accepting a false alarm (α),

2. the probability of missing a target detection (β), and

3. the average test length ($E(T)$).

These three statistics summarize the detection performance of the MMHTT algorithm for different signal models and can be used to evaluate the algorithm's global detection and tracking performance.

4.1 Single Candidate Trajectory Performance

The first stage in analyzing the performance of the MMHTT algorithm is to evaluate the performance of the sequential decision rule used to evaluate the candidate trajectories. The performance of the multistage hypothesis test defined in Chapter 3 can be analyzed for any hypothesized trajectory where the probability distribution of the observations is known. As in [40], the approach of Aroian and Robison [91] will be used to evaluate the resulting detection error probabilities.

The following analysis is a generalization of the analysis in [40] to include the performance of multistage hypothesis tests where the observations are not identically distributed. This allows an evaluation of the performance of a given multistage hypothesis test for candidate trajectories which consist of observations of both the image background and one or more targets. These conditions arise when target trajectories intersect the candidate trajectory for some but not all of its length. The results of this analysis will be used in section 4.2 to analyze the performance of multiple multistage hypothesis tests applied to given target tracking scenarios.

Let the probability distribution function for each test sample X_i be one of

$$\begin{array}{ll} f(x_i \mid H_0): & X_i = \ln L(Y_i) \quad \text{for } Y_i \sim f_Y(y \mid \theta_i \in \theta_{H_0}) \\ f(x_i \mid H_1): & X_i = \ln L(Y_i) \quad \text{for } Y_i \sim f_Y(y \mid \theta_i \in \theta_{H_1}) \end{array} \tag{4.1}$$

and let the corresponding cumulative distribution functions (cdf) be defined as

$$F(x \mid H_j) = \int_{-\infty}^{x} f(t \mid H_j)\, dt,\ j \in \{0, 1\} \tag{4.2}$$

In general, the observed test sample x_i is the log-likelihood ratio of the image observation y_i under the target present and target absent hypotheses

$$x_i = \ln \frac{f(y_i \mid \theta_i \in \theta_{H_1})}{f(y_i \mid \theta_i \in \theta_{H_0})}. \tag{4.3}$$

However, when the image observations y_i are distributed as iid Gaussian random variables ($N(\mu_{H_j}, \sigma^2)$), the truncated SPRT in (3.12) can be expressed as a multistage hypothesis test (2.33) with decision thresholds [40]

$$\begin{array}{lcl} a_i & = & \hat{a}\frac{\sigma^2}{\mu_{H_1} - \mu_{H_0}} + i\frac{\mu_{H_1} + \mu_{H_0}}{2} \\ b_i & = & \hat{b}\frac{\sigma^2}{\mu_{H_1} - \mu_{H_0}} + i\frac{\mu_{H_1} + \mu_{H_0}}{2} \end{array} \tag{4.4}$$

and

$$a_K = \tau \frac{\sigma^2}{\mu_{H_1} - \mu_{H_0}} + K \frac{\mu_{H_1} + \mu_{H_0}}{2} \tag{4.5}$$

For this special case, the test sample x_i is simply the i'th image observation y_i.

The key performance parameters of a multistage hypothesis test for a candidate trajectory with non-iid observations can be defined as:

- the probability of accepting the target absent hypothesis H_0 at stage i,

$$\delta_i^{H_0}(\Theta) \equiv \Pr(\text{accept } H_0 \text{ at stage } i \mid \Theta) \tag{4.6}$$

- the probability of accepting the target present hypothesis H_1 at stage i

$$\delta_i^{H_1}(\Theta) \equiv \Pr(\text{accept } H_1 \text{ at stage } i \mid \Theta) \tag{4.7}$$

- and the probability that the test reaches stage $i + 1$.

$$\delta_i^{H_\emptyset}(\Theta) \equiv \Pr(\text{test reaches stage } i + 1 \mid \Theta) \tag{4.8}$$

where the distribution of the image observations $\mathbf{Y} = \{Y_1, \ldots, Y_n\}$, along the candidate trajectory, are defined by a sequence of parameter vectors

$$\Theta \equiv \{\theta_j \mid Y_j \sim f_Y(y \mid \theta_j),\ 1 \leq j \leq n\} \tag{4.9}$$

indicating the sequence of target present and target absent image observations. Thus, the performance of a multistage hypothesis test (2.33) along an arbitrary candidate trajectory can be analyzed as follows:

Define the pdf of the i'th test sample X_i as

$$f_i(x) = f(x_i \mid H_j),\ j \in \{0, 1\} \tag{4.10}$$

the corresponding cdf as

$$F_i(x) = F(x_i \mid H_j),\ j \in \{0, 1\} \tag{4.11}$$

and the pdf of the cumulative test statistic $\sum_{j=1}^{i} x_j$ as

$$f_i^\star(x) \tag{4.12}$$

Since the observations $\mathbf{y}$ were assumed to be independent, the test samples $\mathbf{x} = \{x_1, \ldots, x_n\}$ are independent and the probability distribution $f^\star_{i+1}(x)$ can be computed as

$$f^\star_{i+1}(w) = \int_{-\infty}^{\infty} f_{i+1}(w-x) f^\star_i(x) dx \tag{4.13}$$

where $w = x_{i+1} + \sum_{j=1}^{i} x_j$ [92]. Taking into consideration that a multistage hypothesis test will continue if and only if $b_i \leq \sum_{j=1}^{i} x_j \leq a_i$, define

$$\begin{aligned} \hat{f}_1(x) &= f_1(x) \\ \hat{f}_{i+1}(w) &= \begin{cases} \int_{b_i}^{a_i} f_{i+1}(w-x)\hat{f}_i(x)dx & \text{if } b_i \leq \sum_{j=1}^{i} x_j \leq a_i \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.14}$$

Then the first stage performance expressions are easily obtained as

$$\delta_1^{H_0}(\Theta) = F_1(b_1) \tag{4.15}$$

$$\delta_1^{H_1}(\Theta) = 1 - F_1(a_1) \tag{4.16}$$

$$\delta_1^{H_\emptyset}(\Theta) = \int_{b_1}^{a_1} f_1(x)\, dx \tag{4.17}$$

Thus, the probability that the test continues after stage $i+1$ (i.e. reaches stage $i+2$), is given by

$$\delta_{i+1}^{H_\emptyset}(\Theta) = \int_{b_{i+1}}^{a_{i+1}} \hat{f}_{i+1}(x)\, dx \tag{4.18}$$

the probability of accepting H_0 at stage $i+1$ is

$$\begin{aligned} \delta_{i+1}^{H_0}(\Theta) &= \int_{-\infty}^{b_{i+1}} \hat{f}_{i+1}(x)dx \\ &= \int_{b_i}^{a_i} F_{i+1}(b_{i+1}-x))\hat{f}_i(x)\, dx \end{aligned} \tag{4.19}$$

and the probability of accepting H_1 at stage $i+1$ is

$$\begin{aligned} \delta_{i+1}^{H_1}(\Theta) &= \int_{a_{i+1}}^{\infty} \hat{f}_{i+1}(x)dx \\ &= \int_{b_i}^{a_i} (1 - F_{i+1}(a_{i+1}-x))\hat{f}_i(x)\, dx \end{aligned} \tag{4.20}$$

where we have used the fact that

$$\int_{-\infty}^{a_{i+1}} \hat{f}_{i+1}(t)\, dt = \int_{-\infty}^{a_{i+1}} \int_{b_i}^{a_i} f_{i+1}(t-x)\hat{f}_i(x)\, dx\, dt \tag{4.21}$$

$$= \int_{b_i}^{a_i} \int_{-\infty}^{a_{i+1}} f_{i+1}(t-x)\hat{f}_i(x)\, dt\, dx \tag{4.22}$$

$$= \int_{b_i}^{a_i} F_{i+1}(a_{i+1}-x)\hat{f}_i(x)\, dx. \tag{4.23}$$

These expressions can be evaluated using standard numerical integration techniques as long as $f(x \mid H_j)$ is sufficiently smooth [93].

Since the test must terminate in exactly one of the K stages, the power function consists of K mutually exclusive events. Thus,

$$\Pr(\text{accept } H_1 \text{ at any stage} \mid \Theta) = \sum_{i=1}^{K} \delta_i^{H_1}(\Theta), \tag{4.24}$$

the false alarm rate for any hypothesized trajectory is

$$\alpha_{HT}(\Theta) = \sum_{i=1}^{K} \delta_i^{H_1}(\Theta), \text{ where } \theta_i \in \theta_{H_0}, \tag{4.25}$$

and the detection probability for any hypothesized trajectory is

$$(1 - \beta_{HT}(\Theta)) = \sum_{i=1}^{K} \delta_i^{H_1}(\Theta), \text{ where } \theta_i \in \theta_{H_1}. \tag{4.26}$$

The average test length, $E(T \mid \Theta)$ is obtained by expressing the probability of terminating in the i^{th} stage as $\delta_{i-1}^{H_\emptyset}(\Theta) - \delta_i^{H_\emptyset}(\Theta)$, for $1 \leq i \leq K$, and the probability of terminating in the final stage as $\delta_{K-1}^{H_\emptyset}(\Theta)$. Thus, define $\delta_0^{H_\emptyset}(\Theta) \equiv 1$, and

$$\begin{array}{ll} E(T \mid \Theta) & = [\sum_{i=1}^{K-1} i(\delta_{i-1}^{H_\emptyset}(\Theta) - \delta_i^{H_\emptyset}(\Theta))] + K\delta_{K-1}^{H_\emptyset}(\Theta) \\ & = \sum_{i=1}^{K} \delta_{i-1}^{H_\emptyset}(\Theta) \end{array}. \tag{4.27}$$

In the next section, these expressions will be used to analyze the detection performance of the MMHTT algorithm for a tree-structured candidate trajectory set.

4.2 Performance for a Candidate Trajectory Tree

The preceding section described a procedure for evaluating the single trajectory detection performance of a multistage hypothesis test, for an arbitrary sequence of observations. Since the target detection algorithm of section 3.1 consists of multiple multistage hypothesis tests, this procedure can be used to evaluate the local detection performance of a candidate trajectory set applied to an arbitrary image sequence. If each of the image observations y_i can be modelled as realizations of random variables from one of two probability distributions, then an evaluation of the stage by stage performance specifications, $\delta_i^H = \{\delta_i^{H_\emptyset}(\Theta), \delta_i^{H_0}(\Theta), \delta_i^{H_1}(\Theta)\}$ for all Θ, is sufficient to bound the detection performance of an arbitrary candidate trajectory set. In general, the analysis of a K-stage candidate trajectory set requires an evaluation of 2^K single trajectory tests. In fact, this analysis can be generalized for image observations which can be modelled as realizations of random variables from one of an arbitrary but finite number of known probability distributions.

If the trajectory of each target in the image sequence is known, the path of each candidate trajectory in the set can be compared to the target trajectories to determine the probability distribution of observations along each candidate trajectory. This sequence can then be used as an index to retrieve the appropriate multistage hypothesis test specifications δ_i^H for each test stage, enabling an analysis of the probability of false alarm, and the probability of missed detection at each stage. It can be shown that the error probabilities for the candidate trajectory set are no greater than the sum of the error probabilities of each candidate trajectory in the set. Thus, summing these results over all the candidate trajectories in the set yields a measure of the total probability of false alarm and the total probability of missed detection for the candidate trajectory set.

Figure 4.1: A 3-Stage Candidate Trajectory Tree.

As an example, consider the K=3-stage candidate trajectory tree depicted in figure 4.1. This tree contains 49 candidate trajectories covering the complete range of discrete target trajectories with a maximum target velocity of 1 pixel/frame. Each candidate trajectory originates in the root pixel $(0, 0)$ of the first frame and terminates in one of the 25 pixels in the velocity annulus of the third frame (see Table 4.1).

Target Pixel Locations for 3-stage Candidate Trajectory Tree							
Candidate	Image Frame			Candidate	Image Frame		
Trajectory	1	2	3	Trajectory	1	2	3
1	(0,0)	(0,0)	(0,0)	26	(0,0)	(-1,0)	(-2,-1)
2	(0,0)	(0,0)	(-1,1)	27	(0,0)	(-1,0)	(-2,0)
3	(0,0)	(0,0)	(0,1)	28	(0,0)	(-1,0)	(-1,-1)
4	(0,0)	(0,0)	(-1,-1)	29	(0,0)	(-1,0)	(-1,0)
5	(0,0)	(0,0)	(-1,0)	30	(0,0)	(0,-1)	(-1,-2)
6	(0,0)	(0,0)	(0,-1)	31	(0,0)	(0,-1)	(-1,-1)
7	(0,0)	(0,0)	(1,-1)	32	(0,0)	(0,-1)	(0,-2)
8	(0,0)	(0,0)	(1,0)	33	(0,0)	(0,-1)	(0,-1)
9	(0,0)	(0,0)	(1,1)	34	(0,0)	(0,-1)	(1,-2)
10	(0,0)	(-1,1)	(-2,1)	35	(0,0)	(0,-1)	(1,-1)
11	(0,0)	(-1,1)	(-2,2)	36	(0,0)	(1,-1)	(1,-2)
12	(0,0)	(-1,1)	(-1,1)	37	(0,0)	(1,-1)	(1,-1)
13	(0,0)	(-1,1)	(-1,2)	38	(0,0)	(1,-1)	(2,-2)
14	(0,0)	(0,1)	(-1,1)	39	(0,0)	(1,-1)	(2,-1)
15	(0,0)	(0,1)	(-1,2)	40	(0,0)	(1,0)	(1,-1)
16	(0,0)	(0,1)	(0,1)	41	(0,0)	(1,0)	(2,-1)
17	(0,0)	(0,1)	(0,2)	42	(0,0)	(1,0)	(1,0)
18	(0,0)	(0,1)	(1,1)	43	(0,0)	(1,0)	(1,1)
19	(0,0)	(0,1)	(1,2)	44	(0,0)	(1,0)	(2,0)
20	(0,0)	(-1,-1)	(-2,-2)	45	(0,0)	(1,0)	(2,1)
21	(0,0)	(-1,-1)	(-2,-1)	46	(0,0)	(1,1)	(1,1)
22	(0,0)	(-1,-1)	(-1,-2)	47	(0,0)	(1,1)	(1,2)
23	(0,0)	(-1,-1)	(-1,-1)	48	(0,0)	(1,1)	(2,1)
24	(0,0)	(-1,0)	(-2,1)	49	(0,0)	(1,1)	(2,2)
25	(0,0)	(-1,0)	(-1,1)				

Table 4.1: Candidate Trajectories for a 3-Stage Tree

Assuming that two probability distributions are sufficient to model the image observations in the presence and absence of a target, the detection performance of each candidate trajectory can be described by one of the $2^3 = 8$ single trajectory analyses depicted in figure 4.2 and listed in Table 4.2. Note that the performance statistics for the early stages of the test are shared by several trajectory analyses. In general, the i^{th} stage performance statistics are shared by 2^{K-i} other single trajectory analyses.

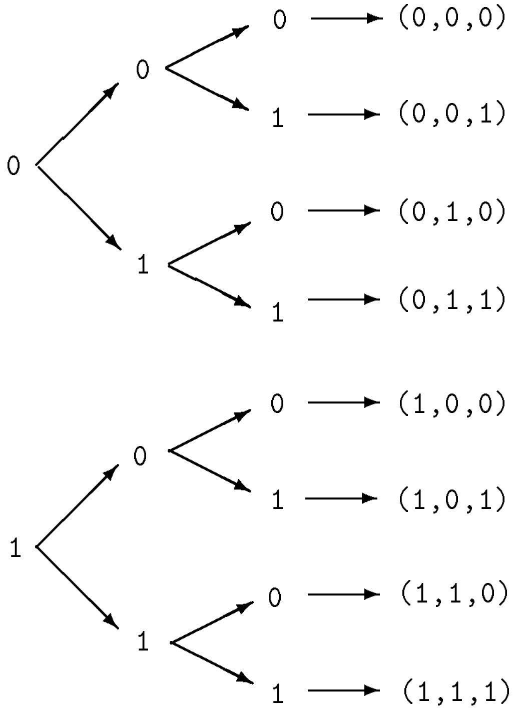

Figure 4.2: Performance Analysis Tree ($K = 3$)

For a known target trajectory, each of the 49 candidate trajectories in Table 4.1 has a sequence of target present and target absent observations corresponding with one of the trajectory analyses in Table 4.4. For example, if it is known that the true target trajectory has coordinates {(0,0), (1,0), (2,0)} relative to the origin of the candidate trajectory set, then the candidate trajectories in Table 4.1 can be mapped to the performance analyses listed in Table 4.4 (see Table 4.3). Note that if candidate trajectory 44 is the desired target trajectory, candidate trajectories $\{1, 2, \ldots, 39, 46, 47, 48, 49\}$

Stage	Theta	δ^H
1	$\Theta_0 = \{\theta_{H_0}\}$	$\delta_1^H(\Theta_0)$
	$\Theta_1 = \{\theta_{H_1}\}$	$\delta_1^H(\Theta_1)$
2	$\Theta_2 = \{\theta_{H_0}, \theta_{H_0}\}$	$\delta_2^H(\Theta_2)$
	$\Theta_3 = \{\theta_{H_0}, \theta_{H_1}\}$	$\delta_2^H(\Theta_3)$
	$\Theta_4 = \{\theta_{H_1}, \theta_{H_0}\}$	$\delta_2^H(\Theta_4)$
	$\Theta_5 = \{\theta_{H_1}, \theta_{H_1}\}$	$\delta_2^H(\Theta_5)$
3	$\Theta_6 = \{\theta_{H_0}, \theta_{H_0}, \theta_{H_0}\}$	$\delta_3^H(\Theta_6)$
	$\Theta_7 = \{\theta_{H_0}, \theta_{H_0}, \theta_{H_1}\}$	$\delta_3^H(\Theta_7)$
	$\Theta_8 = \{\theta_{H_0}, \theta_{H_1}, \theta_{H_0}\}$	$\delta_3^H(\Theta_8)$
	$\Theta_9 = \{\theta_{H_0}, \theta_{H_1}, \theta_{H_1}\}$	$\delta_3^H(\Theta_9)$
	$\Theta_{10} = \{\theta_{H_1}, \theta_{H_0}, \theta_{H_0}\}$	$\delta_3^H(\Theta_{10})$
	$\Theta_{11} = \{\theta_{H_1}, \theta_{H_0}, \theta_{H_1}\}$	$\delta_3^H(\Theta_{11})$
	$\Theta_{12} = \{\theta_{H_1}, \theta_{H_1}, \theta_{H_0}\}$	$\delta_3^H(\Theta_{12})$
	$\Theta_{13} = \{\theta_{H_1}, \theta_{H_1}, \theta_{H_1}\}$	$\delta_3^H(\Theta_{13})$

Table 4.2: Performance Tree Analysis Nodes ($K = 3$)

contain no target observations and candidate trajectories $\{40, 41, 42, 43, 45\}$ only differ from the true target trajectory in the last stage of the multistage hypothesis test.

	Stage		
Candidate Trajectories	1	2	3
$\{1, 2, \ldots, 39, 46, 47, 48, 49\}$	Θ_1	Θ_4	Θ_{10}
$\{40, 41, 42, 43, 45\}$	Θ_1	Θ_5	Θ_{12}
$\{44\}$	Θ_1	Θ_5	Θ_{13}

Table 4.3: Candidate Trajectory Mapping to Single Track Performance Analyses

	Performance Specifications by Stage		
Θ	Stage 1	Stage 2	Stage 3
Θ_6	$\delta_1^H(\Theta_0)$	$\delta_2^H(\Theta_2)$	$\delta_3^H(\Theta_6)$
Θ_7	$\delta_1^H(\Theta_0)$	$\delta_2^H(\Theta_2)$	$\delta_3^H(\Theta_7)$
Θ_8	$\delta_1^H(\Theta_0)$	$\delta_2^H(\Theta_3)$	$\delta_3^H(\Theta_8)$
Θ_9	$\delta_1^H(\Theta_0)$	$\delta_2^H(\Theta_3)$	$\delta_3^H(\Theta_9)$
Θ_{10}	$\delta_1^H(\Theta_1)$	$\delta_2^H(\Theta_4)$	$\delta_3^H(\Theta_{10})$
Θ_{11}	$\delta_1^H(\Theta_1)$	$\delta_2^H(\Theta_4)$	$\delta_3^H(\Theta_{11})$
Θ_{12}	$\delta_1^H(\Theta_1)$	$\delta_2^H(\Theta_5)$	$\delta_3^H(\Theta_{12})$
Θ_{13}	$\delta_1^H(\Theta_1)$	$\delta_2^H(\Theta_5)$	$\delta_3^H(\Theta_{13})$

Table 4.4: Mixed-Model Performance Analysis ($K = 3$)

For any known target scenario, define $\rho_i(\Theta)$ as the number of distinct candidate trajectories at test stage i, which map to each distinct single trajectory performance analysis Θ_j, $1 \leq j \leq 13$. This quantity is determined by comparing the locations of the candidate trajectory image observations to the known target locations. For example, the mapping of the 3-stage candidate trajectory tree to a target trajectory with coordinates {(0,0),(1,0),(2,0)} yields

$$\begin{array}{llll}
\rho_1(\Theta_0) = 0 & \rho_1(\Theta_1) = 1 & & \\
\rho_2(\Theta_4) = 8 & \rho_2(\Theta_5) = 1 & \rho_2(\Theta_j) = 0 \; j \in \{2, 3\} & \\
\rho_3(\Theta_{10}) = 43 & \rho_3(\Theta_{12}) = 5 & \rho_3(\Theta_{13}) = 1 & \rho_3(\Theta_j) = 0 \text{ for } j \in \{6, 7, 8, 9, 11\}
\end{array} \tag{4.28}$$

If we define the i^{th} stage probability of false alarm for the candidate trajectory set as the probability that the target present hypothesis H_1 is accepted in the absence of a target in the i^{th} stage, for at least one of the candidate trajectories, t_k,

$$\alpha_i = \Pr(\bigcup_{\forall t_k \ni \theta_i \notin \theta_{H_1}} \text{accept } H_1 \text{ at stage } i). \tag{4.29}$$

Using Bonferroni's inequality [92],

$$\alpha_i \leq \sum_{j=2^i-2}^{2^{i+1}-3} \rho_i(\Theta_j)\delta_i^{H_1}(\Theta_j),\ \theta_i \in \theta_{H_0}. \tag{4.30}$$

Similarly, the probability of missed detection β_i for each stage of the candidate trajectory is bounded as

$$\beta_i \leq \sum_{j=2^i-2}^{2^{i+1}-3} \rho_i(\Theta_j)\delta_i^{H_0}(\Theta_j),\ \theta_i \in \theta_{H_1}. \tag{4.31}$$

Alternatively, α_i and β_i could be evaluated directly. However, although the individual image observations are mutually independent, the candidate trajectories share at least one common observation (the root pixel), and the i^{th} stage acceptances of a given hypothesis for any two candidate trajectories are dependent events. Thus, an explicit evaluation of α_i and β_i requires a multidimensional integration of the joint probability of the image observations, which is impractical for more than a few image observations.

The detection performance of a sample 3-stage test for Gaussian distributed observations with

$$\begin{aligned} H_1 :&\quad Y_i \sim N(4.0, 1.0) \\ H_0 :&\quad Y_i \sim N(0.0, 1.0) \end{aligned} \tag{4.32}$$

is listed in Table 4.5. The probabilities of accepting hypotheses $H_\emptyset$, H_0 and H_1, for each candidate trajectory, at each stage of the multistage hypothesis test, are retrieved from Table 4.5 using the appropriate index Θ_i from Table 4.3. These results are summarized in Table 4.6.

Summing these probabilities over all the candidate trajectories in the candidate trajectory set, upper bounds can be obtained for the probability of false alarm α, the probability of detection $1 - \beta$ and the expected number of candidate trajectories η

3-stage Performance Analysis				
Stage	Theta	δ^{H_0}	δ^{H_1}	$\delta^{H_\emptyset}$
1	Θ_0	0.894	2.34e-7	0.105
	Θ_1	0.00299	0.149	0.8475
2	Θ_2	0.0969	3.14e-7	0.00857
	Θ_3	0.00102	0.0120	0.0924
	Θ_4	0.293	0.00256	0.552
	Θ_5	2.37e-4	0.6035	0.244
3	Θ_6	0.00857	5.55e-7	0
	Θ_7	0.00618	0.00239	0
	Θ_8	0.0914	0.00103	0
	Θ_9	0.0184	0.0740	0
	Θ_{10}	0.551	8.63e-4	0
	Θ_{11}	0.275	0.277	0
	Θ_{12}	0.237	0.00699	0
	Θ_{13}	0.0172	0.226	0

Table 4.5: Example Mixed-Model Performance Analysis

Candidate		Stage			
Trajectories		1	2	3	Total
$\{1, 2, \ldots, 39, 46, 47, 48, 49\}$	δ^{H_0}	0.00299	0.293	0.551	0.847
Total: 43	δ^{H_1}	0.149	0.00256	8.63e-4	0.153
$\{40, 41, 42, 43, 45\}$	δ^{H_0}	0.00299	2.37e-4	0.237	0.240
Total: 5	δ^{H_1}	0.149	0.6035	0.00699	0.760
$\{44\}$	δ^{H_0}	0.00299	2.37e-4	0.0172	0.0205
Total: 1	δ^{H_1}	0.149	0.6035	0.226	0.9795

Table 4.6: Candidate Trajectory Mapping to Single Track Performance Analyses

for the candidate trajectory set as a whole (see Table 4.7). Note that although the probability of false alarm for the candidate trajectory set in the absence of targets, α_{NT}, is 3.02×10^{-5} the probability of false alarm for the candidate trajectory set under the assumed target conditions (single target trajectory coincident with candidate trajectory 44), α_{ST}, is 0.09256. This increase in the overall false alarm rate is due to the significant probability of accepting one of {40,41,42,43,45 } as a true candidate trajectory.

Stage	α_{NT}	$1-\beta_{NT}$	α_{ST}	$1-\beta_{ST}$
1	2.34e-7	N/A	0.0	0.149
2	2.82e-6	N/A	0.0205	0.6035
3	2.72e-5	N/A	0.07205	0.226
Total	3.02e-5	N/A	0.09256	0.9785

Table 4.7: Decision Probabilities for Sample Candidate Trajectory Test Set

4.2.1 Comments

In [40], the detection performance of the MSHT algorithm was only evaluated for iid observations. The overall false alarm rate was evaluated for the noise-only case, as

$$\alpha = \sum_{i=1}^{K} \rho_i(\theta_{H_0})\delta_i^{H_1}(\theta_{H_0}). \tag{4.33}$$

The detection performance analysis in [40] has been extended in sections 4.1 and 4.2 to include an evaluation of candidate trajectories with non-iid observations. As demonstrated by the test case (4.32) in Table 4.7, the presence of a single target greatly increases the overall false alarm rate.

This should not be surprising, as the multistage hypothesis test used to evaluate the candidate trajectories was designed to choose between the target present and target absent hypotheses under an assumption of iid observations. This decision was designed to meet specified error probabilities for iid observations under each of the target hypotheses. As demonstrated above, the realized probability of false alarm is greater than the designed value when the assumption of iid observations fails.

The increase in the overall false alarm rate is not a critical weakness of the algorithm. Those false target trajectories with an increased likelihood of being accepted as true target trajectories share common observations with the true target trajectory, and the greater the number of target present observations, the greater the likelihood that the given trajectory will be accepted. Thus, in a sparse target environment, the candidate trajectory sets of individual targets are independent and the increase in false alarms is a local phenomenon limited to the immediate neighbours of the target trajectory pixels.

However, in a dense target environment, the candidate trajectory sets of individual targets are not independent, and the presence of multiple targets in each candidate trajectory set leads to an increase in the detection false alarm rate for each candidate trajectory set. This problem can be analyzed for individual candidate trajectory sets using the analysis procedures discussed above. However, the problem of associating observations to individual targets is more properly analyzed in a target tracking context. In the following section, the analysis procedures for a single candidate trajectory set will be exploited to analyze the tracking performance of the MMHTT algorithm for repeated candidate trajectory sets.

4.3 Overall Detection and Tracking Performance

In the following, the analysis tools developed in sections 4.1 and 4.2 will be used to evaluate several aspects of the MMHTT algorithm's overall detection and tracking performance including the average time to track loss. In addition, the performance analysis developed in section 4.2 will be extended to evaluate the performance of the MMHTT algorithm for an arbitrary known target scenario.

4.3.1 Average Time to Track Loss

An important measure of tracking performance is the average time to track loss. This measure reflects the ability of a system to maintain a target track once it is acquired. We will define track loss, with respect to the MMHTT algorithm, as the event characterized by the rejection of the last candidate trajectory which is coincident with the true target trajectory.

An initial estimate of the average time to track loss can be obtained by analyzing the MMHTT algorithm as a sequence of independent Bernoulli trials. Assuming that the candidate trajectory set is sufficiently dense that at least one candidate trajectory is always coincident with the target trajectory, the MMHTT algorithm can be modelled as a sequence of independent experiments with probability $(1-\beta)$ of detecting the target and probability (β) of missing the target. The expected number of trials before the target is missed can then be obtained as the expected time to the first success in a series of independent Bernoulli trials, where success is defined as missing the target.

It is well known that the expected number of trials needed to obtain the first success is $\frac{1}{p}$, where p is the probability of success [92]. Thus the average number of multistage hypothesis tests prior to track loss can be obtained in terms of the probability of detection for a single candidate trajectory, as

$$E[\# \text{ of tests}] = \frac{1}{\beta}. \tag{4.34}$$

The average time to track loss can then be estimated as the product of the average number of tests prior to track loss and the expected length of a multistage hypothesis test in the presence of a target to obtain

$$E[\text{time to track loss}] = \frac{1}{\beta} E(T \mid H_1). \tag{4.35}$$

A more accurate estimate can be obtained by directly calculating the expected time to track loss for a single candidate trajectory with re-initiation of the multistage hypothesis test when the target is confirmed present. Defining $\rho_{TL}(i)$ as the probability of track loss in the i^{th} frame, the expected time to track loss can be computed as

$$E[\text{time to track loss}] = \sum_{i=1}^{\infty} i \rho_{TL}(i) \tag{4.36}$$

where the probability that the track is lost in the i^{th} frame is the probability that the multistage hypothesis test accepts the target absent hypothesis in the i^{th} frame. A K-stage MSHT can only accept the target absent hypothesis in one of K stages. Thus, in order for the MSHT to accept the target absent hypothesis in stage j, the test must have been initiated in the $(i-j)^{\text{th}}$ frame.

The probability of losing the track in the i^{th} frame can be expressed as

$$\rho_{TL}(i) = \sum_{j=1}^{K} \delta_j^{H_0}(\Theta)\rho_{TI}(i-j), \text{ for } \Theta = \{\theta \mid \theta \in \theta_{H_1}\} \tag{4.37}$$

where $\rho_{TI}(i)$ is defined as the probability of re-initiating the MSHT in the i^{th} frame. Now, consider the development of the target track as a sequence of initiated and confirmed target trajectory segments. The first MSHT initiated in Frame 1, can only terminate in one of K frames including Frame 1. The probability that the test terminates and a new test is initiated can be computed for each frame j, $(1 \leq j \leq K)$, as

$$\delta_j^{H_1}(\Theta), \text{ for } \Theta = \{\theta \mid \theta \in \theta_{H_1}\}. \tag{4.38}$$

For subsequent frames, the probability that the current multistage hypothesis test terminates and a new test is initiated, $\rho_{TI}(i)$ is the probability of accepting the target present hypothesis in the i^{th} frame. A K-stage MSHT can only accept the target present hypothesis in one of K stages. Thus, in order for the MSHT to accept the target present hypothesis in stage j, the test must have been initiated in the $(i-j)^{\text{th}}$ frame. Thus, the probability that a new test is initiated in the i^{th} frame can be recursively computed as

$$\rho_{TI}(i) = \sum_{j=1}^{K} \delta_j^{H_1}(\Theta)\rho_{TI}(i-j) \text{ for } \Theta = \{\theta \mid \theta \in \theta_{H_1}\}. \tag{4.39}$$

Therefore, the expected time to track loss can be recursively estimated as

$$\sum_{i=1}^{N} i \sum_{j=1}^{K} \delta_j^{H_0}(\Theta)\rho_{TI}(i-j), \text{ for } \Theta = \{\theta \mid \theta \in \theta_{H_1}\} \tag{4.40}$$

for large N, where $\rho_{TI}(i)$ is recursively computed by (4.39).

Note that the preceding analysis only evaluated the ability of the MMHTT algorithm to track the target trajectory using a sequence of candidate trajectory segments coincident with the true target trajectory. In practice, as discussed in section 4.2, there is a significant probability of accepting, as a segment of the true target trajectory, those candidate trajectories which differ from the true target trajectory only in the last stage(s) of the MSHT. If the scope of the trajectory search for re-initiated trajectories is not overly strict, there is a significant probability that one of these candidate trajectories will detect the target in some later test stage. Thus, these neighbouring candidate trajectories provide a measure of robustness against track loss and an analysis of repeated candidate trajectory trees suggests an increase in the expected time to track loss.

4.3.2 Performance of Repeated Candidate Trajectory Trees

Although the analysis of a single candidate trajectory or candidate trajectory tree provides an adequate measure of the detection performance of the MMHTT algorithm, an analysis of the dynamics of the repeated MSHT is required for a more detailed analysis of the algorithm's tracking performance. In the following, the performance analysis for a single candidate trajectory tree will be extended to analyze the performance of successive candidate trajectory trees. The following analysis will be restricted to an analysis of the candidate trajectories generated as a result of a single initiated search for new target trajectories. Thus, the analysis will consider the result of successive MSHTs generated by the first test but will not consider the effects of concurrent tests for new targets.

In section 4.2 it was assumed that the trajectory of each target in the image sequence is known and that the image observations y can be modelled as realizations of random variables with one of two known probability distributions. Thus, given an arbitrary set of target trajectories, a reference test target, and an initial position for the initiation of the first candidate trajectory search, the results of section 4.2 can be used to determine the probability that the MMHTT algorithm will accept the target present hypothesis in each node of the candidate trajectory tree. Since a search is initiated for an extension of each confirmed candidate trajectory, the probability of accepting the target present hypothesis at a given node is also the probability that a trajectory search is initiated in that node.

Given that a candidate trajectory search is initiated at (x_i, y_i) in Frame i, the probability that candidate trajectory $t_{\mathbf{x}_i,\mathbf{x}_j}$ from (x_i, y_i) in Frame i to (x_j, y_j) in Frame j is confirmed, can be obtained from the performance analysis of a single candidate trajectory tree.

$$\Pr(\text{accept } t_{\mathbf{x}_i,\mathbf{x}_j} \mid \text{test initiated at } (x_i, y_i) \text{ in Frame } i) = \delta_{j-i}^{H_1}(\Theta) \tag{4.41}$$

where Θ is known:

$$\Theta = \{\theta_i \mid Y_i \sim f_Y(y_i \mid \theta_i),\ \text{for } Y_i \in \mathbf{Y}_{t_{\mathbf{x}_i,\mathbf{x}_j}}\}. \tag{4.42}$$

Since candidate trajectory $t_{\mathbf{x}_i,\mathbf{x}_j}$ can be confirmed if and only if a candidate trajectory search is initiated at (x_i, y_i) in Frame i, the unconditional probability that candidate trajectory $t_{\mathbf{x}_i,\mathbf{x}_j}$ is confirmed can be obtained as

$$\Pr(\text{accept } t_{\mathbf{x}_i,\mathbf{x}_j}) = \delta_{j-i}^{H_1}(\Theta)\Pr(\text{test initiated at } (x_i, y_i) \text{ in Frame } i). \tag{4.43}$$

Note that the probability that a test is initiated at (x_i, y_i) in Frame i is just the probability that one or more of the candidate trajectories from some pixel (x_k, y_k) in a previous Frame to (x_i, y_i) in Frame i, are confirmed.

$$\Pr(\text{test initiated at } (x_i, y_i) \text{ in Frame } i) \leq \iota(\mathbf{x}_i) \equiv \sum_{t_{\mathbf{x}_k,\mathbf{x}_i}} \Pr(\text{accept } t_{\mathbf{x}_k,\mathbf{x}_i}). \tag{4.44}$$

Thus, an upper bound on the probability of initiating a search for a target trajectory extension, at (x_i, y_i) in Frame i, can in principle be computed for every reachable target location.

Given that a test for a new target trajectory is initiated at (x_1, y_1) in Frame 1, $(\iota(\mathbf{x}_1) = 1)$, the probability of initiating a search for an extension of this target trajectory can be computed for each pixel (x_j, y_j) in Frame j of the candidate trajectory search space as

$$\iota(\mathbf{x}_j) = \delta_{j-i+1}^{H_1}(\Theta). \tag{4.45}$$

The probability of initiating a search for a target trajectory extension in subsequent frames can then be computed using (4.44), by evaluating the $\iota(\mathbf{x}_i)$ for each reachable pixel in each subsequent frame. This calculation is computationally impractical for more than a few frames. However, if the analysis is constrained to an evaluation of a set of most probable pixels in each frame, one can still evaluate the dynamic performance of the MMHTT algorithm for an arbitrary target scenario.

Two constraints are useful for bounding the calculation of propagation probabilities for repeated MSHTs. The first constraint is to inhibit the calculation of confirmation probabilities for pixels (x_i, y_i) where $\iota(\mathbf{x}_i)$ is less than a specified error tolerance. The second constraint is to inhibit the calculation of confirmation probabilities for pixels (x_i, y_i) which do not have the test target in their candidate trajectory search space. These constraints limit the evaluation of candidate trajectories to a set of most probable candidates in a neighbourhood of the test target trajectory. For typical test designs, the probability of propagating a candidate trajectory in the absence of target observations is small ($P_{FA} < 10^{-5}$) and the expected track length of a false target trajectory in the image background is much less than the number of stages in the MSHT. Thus, although the candidate trajectories which are ignored by the imposition of these constraints have a finite probability of ocurring, that probability is typically quite small.

Given an estimate of the probability that a trajectory search is initiated for every pixel in a neighbourhood of the test target trajectory, and knowledge of which pixels contain a target, the probability of accepting a false alarm in each image frame can be bounded by the sum of the probabilities of accepting a false alarm in each pixel of the frame:

$$\alpha_i \leq \sum_{\mathbf{x}} \iota(\mathbf{x}), \text{ for each image frame } i. \tag{4.46}$$

Similarly, the probability of detecting any of the known targets in each image frame, is simply the probability that a candidate trajectory is accepted at the target's current pixel location:

$$\Pr(\text{detecting the target at } \mathbf{x_i} \text{ in Frame } i) \leq \iota(\mathbf{x}_i). \tag{4.47}$$

In addition, a rough upper bound for the probability of following the test target can be obtained by constructively computing the probability that at least one of the candidate trajectories at any given stage is coincident with the test target trajectory. Given the position of the test target $\mathbf{x_i}$ in image frame i, the probability that at least one of the candidate trajectories in the algorithm's undecided trajectory list is coincident with the target can be bounded by

$$\sum_{t_{\mathbf{x}_k, \mathbf{x}_i}} \delta_{i-k}^{H_\emptyset}(\Theta)\iota(\mathbf{x}_k) \tag{4.48}$$

In general, this upper bound for the probability of following the target is more optimistic than the measures of the expected time to track loss derived in section 4.3.1. This reflects the robustness to track loss provided by maintaining and evaluating multiple candidate trajectories which are close to the true target trajectory.

4.3.3 A Sample Tracking Performance Analysis

As an example of the target tracking analysis presented in sections 4.3.1 and 4.3.2, the probability of following a point-source target, in a Gaussian noise background, will be analyzed. A 3-stage truncated SPRT, with a nominal detection probability of 0.9, was designed for a constant intensity target with a SNR of 12 dB. An analysis of the resulting test yields an actual probability of detection of 0.9795 and an expected test length, in the presence of a target, of 2.04 image frames.

As discussed in section 4.3.1, an initial estimate of the average time to track loss is given by (4.35),

$$\frac{1}{\beta}E(T \mid H_1) = 98 \text{ frames.} \tag{4.49}$$

A more accurate estimate of 101 image frames is obtained by recursively estimating the expected time to track loss for a single candidate trajectory using (4.40). This is a conservative bound on the overall tracking performance of the MMHTT algorithm. In practice, there is a small but significant probability that, in the event of track loss along the true target trajectory, one of the neighbouring candidate trajectories in the candidate trajectory set will be accepted by the MSHT algorithm *and* the true target trajectory will be detected in a subsequent search for a target trajectory extension.

As discussed in section 4.3.2, a rough upper bound for the probability of following a target can be obtained by analyzing the performance of repeated candidate trajectory trees. The results of an analysis of the probability of following a target, using the 3-stage test described above, are plotted in figure 4.3. The lower curve is the conservative bound obtained by estimating the probability of following a target using (4.40) and the upper curve is the rough upper bound obtained by an analysis of repeated candidate trajectory sets, using (4.48). It was assumed that the target followed a linear trajectory with a constant velocity of 1 pixel/frame and that the candidate trajectory set was designed to evaluate all linear constant velocity trajectories with a maximum velocity of 1 pixel/frame. Note that the actual performance of the MMHTT algorithm is bounded by these two estimates. It is clear that the MMHTT algorithm significantly extends the length of the target trajectories detected by the MSHT algorithm.

Figure 4.3: The probability of following a target

The upper curve is for an analysis of repeated candidate trajectory trees.

The lower curve is for an analysis of a single repeated candidate trajectory.

4.4 Summary

The performance analysis techniques presented in this chapter can only provide a rough measure of the global detection and tracking performance of the MMHTT algorithm for arbitrary target scenarios. The explicit evaluation of better bounds for the algorithm's detection performance or an analysis of the algorithm's computational requirements (e.g. expected number of candidate trajectories per stage) is a difficult problem confounded by the statistical dependence of the candidate trajectories in the candidate trajectory tree. Although the image observations along any single candidate trajectory are independent, the tree-structure of the candidate trajectory set introduces a statistical dependence between candidate trajectories which are children of the same tree node. In principle, exact expressions for the performance measures could be evaluated by computing the joint probability distribution of the image observations. However, the effort involved outweighs its relative value in the design of an appropriate algorithm for a given target scenario.

Several important observations can however be drawn from the analysis presented herin. First, the MMHTT algorithm preserves the detection performance of the MSHT algorithm while extending the length of the detected target trajectories by an order of magnitude for typical test designs. Second, candidate trajectory segments which partially overlap the true target trajectory have a significant probability of being accepted by the MSHT algorithm. Although, these candidate trajectories are, strictly speaking, false alarms they are sufficiently close to the true target trajectory that they provide a measure of robustness to track loss and target manoeuvres. Finally, there is a trade-off between minimizing the probability of missed detection to maximize tracking performance and a minimization of the expected test length in the absence of a target to minimize the computational cost of evaluating the candidate trajectory sets.

Chapter 5

Feature Detection and Tracking

A large number of algorithms, which compute the three-dimensional motion of a rigid object from a set of n feature correspondences over m frames, have been proposed [56,58,94–97]. However, relatively few researchers have addressed the problem of automating the establishment of feature correspondences in a robust and computationally efficient manner. The establishment of feature correspondences is confounded by problems in feature detection which can lead to the disappearance of true features and the appearance of false features in any image frame. Object occlusion, changes in the relative imaging geometry and changes in scene illumination can lead to significant changes in the image feature observations and are a major source of correspondence errors [98].

In the following, a system will be described for automating the establishment of feature correspondences for objects moving against a stationary image background. The proposed system is an application of the MMHTT algorithm, developed in Chapter 3, to the detection and tracking of object feature points over an extended number of image frames. The image preprocessing required to implement the MMHTT algorithm is developed and applied to both real and synthetic image sequences. Finally, the MSHT algorithm is applied to the sample image sequences as a qualitative indication of the potential performance of the MMHTT algorithm.

5.1 Previous Work in Feature Correspondence

The standard approach to the feature correspondence problem is to detect features in each image frame and then address the *correspondence problem* as a matching or *data association* problem [99–103]. These algorithms rely on feature uniqueness and location to resolve the correspondence problem without referring to the underlying motion. If features are missing, false features detected, or the distance between distinct feature points is comparable to the expected feature displacement in subsequent frames, the two-frame correspondence problem can be ill-posed.

Consequently, multiframe algorithms have been developed which exploit motion continuity to solve the correspondence problem for temporally dense image sequences [104–107]. These algorithms assume that the motion of imaged objects cannot change instantaneously and thus, if the image sampling rate is sufficiently high, the motion of the projected features on the image plane is *smooth* [104]. However, in an image sequence with a dense set of features and/or in the presence of false feature detections, the search for the smoothest path may lead to significant correspondence errors. Under these conditions the motion smoothness constraint is insufficient to correctly resolve the correspondence problem.

Feature detection and correspondence in a dense set of features has been addressed by applying the MSHT algorithm to detect image features along short, linear trajectories through several images [60, 108]. Although this multiframe detection approach offers increased robustness to feature detection problems and provides implicit local correspondence decisions, it fails to extend the detected feature trajectories. Consequently, a heuristic feature path linking algorithm was proposed to produce the long trajectories required for multiframe estimation of structure and motion.

Recently, Chang and Aggarwal identified similarities between the correspondence problem and the multitarget tracking data association problem. They posed the feature correspondence problem as a joint problem in feature detection and estimation [59]. Their algorithm combines statistical detection of two-dimensional line features in the image plane with an application of the JPDA filter to estimate the structure and motion of three-dimensional lines [59].

5.2 The Feature Point Correspondence Problem

The feature detection and correspondence problems are fundamentally decision problems. The detection problem is a decision between the two hypotheses

H_1: $I[x, y, t]$ is a feature point, and

H_0: $I[x, y, t]$ is not a feature point

and the correspondence problem is a decision between the two hypotheses

H_1: $I[x, y, t]$ and $I[x', y', t']$ are observations of the same object feature, and

H_0: $I[x, y, t]$ and $I[x', y', t']$ are not observations of the same object feature

The approach proposed herein is to unify the feature detection and correspondence problems in a common decision framework using the MMHTT algorithm.

It is assumed that the imaged objects have surface markings whose projections on the image plane are identifiable, whose locations can be accurately determined, and that after appropriate image preprocessing the image pixel intensities can be modelled as realizations of independent random variables from one of two probability distributions

$$\begin{array}{ll} f_1(I[x, y, t]) & \text{if the pixel is a feature point, or} \\ f_0(I[x, y, t]) & \text{otherwise.} \end{array}$$

Thus, the multiframe feature point detection and tracking problem can be defined analogously to the MFTDT problem of Chapter 1.

Figure 5.1: Automated Feature Detection and Tracking System

The remainder of this chapter is dedicated to the development of a signal and data processing structure for automating the establishment of feature point correspondences in a sequence of digital images (see figure 5.1). This solution to the correspondence problem builds on the work in [60], by replacing the original implementation of the MSHT algorithm with the MMHTT algorithm developed in Chapter 3. The MMHTT algorithm enables the detection *and* tracking of feature point correspondences over an extended number of image frames, obviating the need for the heuristic feature path linking step in [60]. The resulting system combines the detection power and efficiency of the MSHT algorithm with the well-documented error performance of a MHT approach to data association.

As in [60], the system described in this chapter is designed to detect and track local extrema of the image intensity surface with a high Gaussian curvature. The system accepts an image sequence with a stationary image background as input and provides a set of feature trajectories as output. Each feature trajectory consists of a set of feature positions in a subset of the image sequence.

As discussed in Chapter 1, this system structure can be divided into three processing tasks:

1. image preprocessing,
2. feature detection, and
3. feature tracking.

The initial preprocessing is designed to compute features and optimize usage of the system's limited computational resources. The remaining tasks are satisfied by an implementation of the MMHTT algorithm developed in Chapter 3. A sample of the first two stages of the system output, for both real and synthetic images, is provided to illustrate the potential for an application of the MMHTT algorithm.

5.3 Image Preprocessing

The goals of the initial image sequence preprocessing are to compute potentially trackable features and detect regions of significant inter-frame change. This is a data reduction step designed to optimize usage of the feature tracker's limited computational resources. These functions are applied to single images and are typically suitable for implementation on special purpose hardware which can easily implement these functions at the standard video frame rate.

5.3.1 Features

The key to a feature-based approach to image sequence analysis is the computation or detection of appropriate features. The selection of appropriate features is both task and data dependent. However, there are several design criteria which should be considered.

Feature Selection Criteria:

1. Discriminability: The features should be designed to enhance detectability.
2. Localizability: It should be possible to accurately determine feature positions in the image.
3. Robustness: Selected features should be robust to changes in the imaging environment.
4. Sparseness: Selected features should be sparsely distributed to reduce track crosscorrelation.
5. Computability: Feature maps should be simple to compute.

As an illustration of the MMHTT approach to the feature correspondence problem, the system described herein is designed to detect and track the Gaussian curvature features proposed in [60]. These features are critical points of the image intensity surface with a high Gaussian curvature. It will be shown that these features are simple to compute and that their position is well defined.

Computing Gaussian Curvature Features

Gaussian curvature feature points are local extrema of the image intensity surface with a high Gaussian curvature. Gaussian curvature is a parametrization invariant measure of surface curvature defined as the product of the principal normal curvatures [109]. Thus, computing these features is a two-step process. First, the local intensity extrema are located and then the Gaussian curvature of the surface is evaluated.

The local intensity extrema are estimated by the extrema of a parametric surface fit to the image intensity data in a local neighbourhood. The image intensities are fit to a parabolic surface, because it has a unique critical point and a minimal number of defining parameters. Although the parabolic surface is a crude fit to the image intensity surface, it captures the gross characteristics of the surface and is more robust to small fluctuations in the value of any single pixel than a higher order model.

Given a local $(2n+1) \times (2n+1)$ neighbourhood of pixels centred on (x, y)

$$\{I[x+i, y+j] \mid |i| \leq n, |j| \leq n\} \tag{5.1}$$

the image intensity surface can be fit to a parabolic surface

$$z = ax^2 + bxy + cy^2 + dx + ey + f \tag{5.2}$$

by finding the coefficients a, b, c, d, e, f, which minimize

$$\|\mathbf{Ax} - \mathbf{G}\|^2 = \sum_{i=-n}^{n} \sum_{j=-n}^{n} (ai^2 + bij + cj^2 + di + ej + f - I(x+i, y+j))^2 \tag{5.3}$$

where $\mathbf{x} = [a\, b\, c\, d\, e\, f]^T$, the k'th row of $\mathbf{A}$ is $[i^2 \; ij \; j^2 \; i \; j \; 1]$, and (i, j) is given by the k'th element of $\mathbf{G}$: $I[x+i, y+j]$. This is a standard linear least squares problem with solution

$$\mathbf{x} = (\mathbf{A^T A})^{-1} \mathbf{A^T G}. \tag{5.4}$$

Since $\mathbf{A}$ is independent of the image intensities, it can be precomputed for a particular neighbourhood. The 6 surface parameters a, b, c, d, e, f, can thus be obtained from the output of 6 FIR filters applied to the image neighbourhood.

Given the parabolic surface parameters, the next problem is to find the extrema of the surface curvature. A standard theorem in multivariate calculus states that a point (x_0, y_0) is a strict local minimum of z if the following conditions are satisfied [110]:

1. $\frac{\delta z}{\delta x}(x_0, y_0) = \frac{\delta z}{\delta y}(x_0, y_0) = 0$, and

2. the Hessian matrix $\begin{bmatrix} \frac{\delta^2 z}{\delta x^2} & \frac{\delta^2 z}{\delta y \delta x} \\ \frac{\delta^2 z}{\delta x \delta y} & \frac{\delta^2 z}{\delta y^2} \end{bmatrix}$ is positive definite.

The first condition is satisfied by $(i, j) = (i_0, j_0)$, where

$$\begin{aligned} i_0 &= \tfrac{2cd-be}{b^2-4ac}, \text{and} \\ j_0 &= \tfrac{2ae-bd}{b^2-4ac} \end{aligned} \tag{5.5}$$

for a non-degenerative surface. Surfaces for which $b^2 - 4ac = 0$ are considered degenerate and have non-unique critical points (i.e. a fold: the extrema points form a line). Reparametrizing the parabolic surface so the extrema is at the origin yields

$$z = ai'^2 + bi'j' + cj'^2 + f', \tag{5.6}$$

where

$$f' = \frac{cd^2 - bde + ae^2 + b^2 f - 4acf}{b^2 - 4ac}. \tag{5.7}$$

The Hessian matrix $\mathbf{H}$ is now a function of three parameters:

$$\mathbf{H} = \begin{bmatrix} 2a & b \\ b & 2c \end{bmatrix} \tag{5.8}$$

The Hessian is positive definite if and only if

1. $a > 0$, and

2. $\det \mathbf{H} = 4ac - b^2 > 0$

Similar theorems state that $(x + i_0, y + j_0)$ is a strict local maximum if and only if

1. $a < 0$, and

2. $\det \mathbf{H} = 4ac - b^2 > 0$

and $(x + i_0, y + j_0)$ is a *saddle point* if $\det \mathbf{H} < 0$. Note that the determinant of the Hessian matrix is zero for a degenerate surface implying that the surface is a function of a single variable.

The Gaussian curvature of a surface is the product of the principal normal curvatures of the surface at any point (x_0, y_0) on the surface. The principal normal curvatures at a point are the eigenvalues of the Hessian matrix evaluated at that point. Thus, the Gaussian curvature is equal to the product of the eigenvalues of the Hessian matrix or by a standard theorem of linear algebra, equal to the determinant of the Hessian matrix $\mathbf{H}$.

Thus, every image in the sequence can be processed for Gaussian curvature features as a simple function of the output of six digital filters. These filters compute the surface parameters of parabolic surface patches fit to the image intensities in local neighbourhoods of the image. The remaining problem is how to generate a feature image suitable for input to the MMHTT algorithm.

In [60], Debrunner proposed the generation of two *feature maps* to be processed in parallel. One feature map consists of local intensity maxima with high Gaussian curvature and the second feature map consists of local intensity minima with high Gaussian curvature. Each feature map is generated as follows:

1. A parabolic surface is fit to a neighbourhood of each interior pixel in the image.
2. The location of the fitted surface critical point is computed as a simple function of the fitted surface parameters.
3. The Hessian of the fitted surface is evaluated at the critical point.
4. If the Hessian is positive definite or negative definite, the Gaussian curvature K is computed.
5. the value of the $(x + i_0, y + j_0)$ pixel in the appropriate feature map is then incremented by $\frac{K}{1+i_0^2+j_0^2}$.

Note that the contribution of an image neighbourhood to the feature map decays as a function of the inter-pixel distance between the centre of the fitted surface and the centre of the pixel containing the surface extremum. Thus, each feature map will have pixels with a high intensity where the surface fit has a large Gaussian curvature and low intensity pixels in regions with less pronounced variations in image intensity. In practice, this procedure isolates local intensity peaks and corners which may be due to the physical reflectivity of the imaged objects, as desired, or due to spurious photometric effects (e.g. shadows and specular reflections).

5.3.2 Change Detection

The second component of image preprocessing for feature detection and tracking is the detection of regions of significant inter-frame change. This is an auxiliary region of interest process, as discussed in section 3.3.1, which reduces the computational cost of feature detection and tracking by identifying regions of each image which are most likely to contain the moving objects of interest. Such a process could be implemented in parallel with the computation of image feature maps for a real-time implementation.

In the following, a change detector is used to separate moving targets from the stationary background. Ideally, the change detector should be robust to camera noise and changes in illumination. Previous change detection algorithms based on frame differences are sensitive to noise and yield a poor segmentation [111, 112]. Other approaches based on adaptive reference images require a good initial estimate of the stationary background [112, 113]. However, a recent adaptive technique developed by Karmann et al. [111] can cope with illumination changes, system initialization in the presence of moving objects and previously stationary objects which begin to move. This technique generates an adaptive reference image from the image sequence which captures the stationary image background and enhances the detectability of moving objects.

The following is a brief outline of Karmann's technique [113]:

Significant inter-frame changes are detected by comparing the absolute grey level difference of an incoming image frame and an adaptive reference image to a noise-adaptive threshold, T_{na}.

$$\begin{aligned} ChangeImage[x][y] &= 1, \text{ if } |I[\mathbf{x},t] - R[\mathbf{x},t]| > T_{na} \\ &= 0, \text{ otherwise} \end{aligned} \tag{5.9}$$

The current reference image, $R(\mathbf{x},t)$, is estimated from the image sequence using a simple Kalman filter.

$$R[\mathbf{x},t+1] = R[\mathbf{x},t] + K[\mathbf{x},t](I[\mathbf{x},t] - R[\mathbf{x},t]) \tag{5.10}$$

where the Kalman gain $K[\mathbf{x}, t]$ is computed as

$$K[\mathbf{x}, t] = \frac{V[\mathbf{x}, t]}{(I[\mathbf{x}, t] - R[\mathbf{x}, t])^2 + V[\mathbf{x}, t]} \tag{5.11}$$

and $V[\mathbf{x}, t]$ is a measure of the local image noise power estimated by

$$V[\mathbf{x}, t+1] = (1 - K[\mathbf{x}, t]) + S[\mathbf{x}, t] \tag{5.12}$$

with system noise variance $S[\mathbf{x}, t]$.

Following Donohoe [112], the decision threshold T_{na} is derived from the classical likelihood ratio detector,

$$\frac{\Pr(I[\mathbf{x}, t] \mid H_1)}{\Pr(I[\mathbf{x}, t] \mid H_0)} \begin{cases} \geq T_{na} & \Rightarrow \text{choose } H_1 \\ < T_{na} & \Rightarrow \text{choose } H_0 \end{cases} \tag{5.13}$$

which for uniformly distributed target intensities and Gaussian noise statistics, yields,

$$T_{na} = \sqrt{2V \ln \frac{\frac{k}{\lambda}}{\sqrt{2\pi V}}} \tag{5.14}$$

where k is the number of image grey levels, λ is the *a priori* probability ratio of target pixels to background pixels, and V is the average of $V[\mathbf{x}, t]$ over the entire image.

5.4 Feature Detection and Tracking Experiments

In the following, a potential application of the MMHTT algorithm to the feature detection and tracking problem will be investigated. In principle, the algorithm could be designed to detect certain image features directly from the raw image sequence data and/or multiple feature types could be tracked by parallel versions of the algorithm. However, in the following it will be assumed that the feature map is obtained from the Gaussian curvature filter described in the previous section and that the MMHTT algorithm is applied to each feature map separately.

The MSHT algorithm was applied to the feature detection and tracking problem in [60]. However, as discussed previously, the short, detected feature paths are insufficient for motion estimation. Consequently, a heuristic feature path linking algorithm was proposed in [60] to form extended feature trajectories from the short, linear path segments detected by the MSHT algorithm. In the following, the MMHTT algorithm is proposed as a replacement for the MSHT algorithm to both detect and track image features over an extended number of image frames.

As discussed in the prelude to Chapter 3, the MMHTT algorithm assumes that the input image observations, after preprocessing, satisfy (2.12) with independent observations $\mathbf{y}$ such that

$$f_{\mathbf{Y}}(\mathbf{y} \mid \theta) = \prod_{y \in \mathbf{y}} f_{\mathbf{Y}}(y \mid \theta) \tag{5.15}$$

and that the target or feature point satisfies the point-source target model of (2.3). Thus, the first task in designing an appropriate MMHTT algorithm for the detection and tracking of the Gaussian curvature features described in section 5.3.1 is to ascertain the probability distribution of the image observations conditioned on each of the target hypotheses H_0 and H_1.

In [60], the Gaussian curvature feature map observations were modelled as Gaussian random variables with a uniform variance σ^2

$$\begin{aligned} H_1 &: \mathbf{Y} \sim N(\mu, \sigma^2) \\ H_0 &: \mathbf{Y} \sim N(0, \sigma^2). \end{aligned} \tag{5.16}$$

This choice was motivated by a desire to utilize the test threshold design procedures in [40] and Debrunner admits that the statistical properties of the Gaussian curvature feature maps are not truly in accordance with the model. However, reasonable results were obtained in [60] by estimating σ^2 with an empirically determined noise variance for the feature observations. This approach will be maintained in the following experiments.

The results presented herein are not exact quantitative measures of the algorithm's tracking performance. Rather, they are a qualitative indication of the potential for tracking Gaussian curvature features with the MMHTT algorithm in three different imaging environments. Unfortunately, a true qualitative appreciation of the results is only possible in a dynamic medium. However, the results presented here do demonstrate the potential for an application of the MMHTT algorithm to detecting feature trajectories.

In the following, results will be described for one synthetic and two real image sequences. The individual frames of the synthetic sequence were rendered using the ray-tracing software RADIANCE [114]. The object animation in the synthetic sequence is the result of an animation driver for RADIANCE developed by the author, using a motion model described by Young and Chellappa [115]. The video image sequences were digitized at 60 fields/sec., using a Panasonic WV-BD400 CCD video camera.

Four sample images from each sequence are presented. The images have a 256×256 pixel resolution with 255 grey levels. Each image is a sample of the results after a different stage in the sequence of processing steps.

- Top Left: Image Frame of the Original Sequence.
- Top Right: Feature Image Frame.
- Bottom Left: Image Frame After Change Detection.
- Bottom Right: Final Set of Detected Candidate Trajectories.

5.4.1 Synthetic Image Sequence

The synthetic sequence in Images (1-4) depicts a rotating box in a synthetic lab with fluorescent lights. The box was tumbling freely in space with a constant translational acceleration and a rotation with constant precession. The feature frame in Image 2 clearly shows the extraction of curvature features along the edges of the 'real' box and the 'reflection' box in the floor. Features along the edges of the walls and the lights are also extracted.

The background of the synthetic sequence is easily obtained by rendering one frame without the box. This corresponds with the case where a good reference frame is available for the change detector. If the change detector can observe the stationary background environment before moving targets enter the field of view, it greatly enhances detection performance. Note that the change detector correctly segments the moving box from the background without error.

In the final frame, Image 4, the paths of the detected trajectories are displayed. The algorithm correctly tracks the corners of the tumbling box. This performance is

more evident in a dynamic medium where one can observe the overlaid trajectories following the corners of the box in the original sequence.

5.4.2 Real Image Sequences

The first real image sequence is of a translating stack of Sun workstation boxes with the logo clearly visible. This sequence was taken at night with the CCD camera described previously. Contrary to expectations, the lighting conditions in the laboratory, with fluorescent ceiling illumination at night, were not homogeneous. The extracted feature frames for the real sequences are much denser than those in the synthetic case. Again, the curvature features appear along distinct physical and contrast edges and corners in the image sequence. Particularly strong features are extracted for the top corners of the Sun box and along the arm of the demonstrator.

In this sequence, the change detector was not allowed a previous view of the stationary environment. However, it successfully adapted to the moving box and the reference image rapidly converged to the true background. The moving Sun boxes and the demonstrator's arm were clearly segmented. The change detector did not isolate the long box used to push the Sun boxes. In the final frame, Image 8, the detected trajectories for the corners of the Sun boxes and various points along the demonstrator's arm and the Sun logo clearly indicate a translational motion.

The second image sequence is an extremely challenging one to analyze. The sequence was taken during the day and a puddle of sunlight from a window just to the right of the image is clearly visible on the floor beneath the chair. The chair was translating and rotating as it moved from right to left. The rapid rotation of the chair led to the disappearance of most of the curvature features on the surface of the chair. The chair itself presents smooth, textured surfaces and thus does not have any easily detectable Gaussian curvature features. The rapid changes in illumination, poor contrast and lack of strong features in this image provide significant challenges to a feature tracking algorithm.

The features in Image 10 are much harder to detect than those in Image 2 and Image 6. In fact, the chair is barely discernible in Image 10. Strong features are however detected in the stationary image background. Again, the change detector

was not switched on during the object's motion and was challenged to compensate for the moving object and the rapid changes in illumination. The change detector successfully isolated the moving chair, although it took much longer for the reference image to converge towards the true background. The effects of the sunlight are also visible as large vertical bands across the chair in Image 11. Some of the low observable features were tracked and their final trajectories are displayed in Image 12. This sequence demonstrates the algorithm's ability to cope with missed detection. Although the features on the surface of the chair were weak, the algorithm managed to track parts of the chair across the room.

5.5 Summary

The experiments presented in the preceding section provide a qualitative indication of the potential of the MMHTT algorithm for detecting and tracking the Gaussian curvature features proposed in [60]. The success of the MSHT algorithm in detecting short, feature paths suggests that an application of the MMHTT algorithm could improve upon the performance of the MSHT algorithm in [60] by extending the detected feature paths. The introduction of the change detection process in section 5.3.2, as an auxiliary region of interest process (see section 3.3.1) improves the performance of either the MSHT or the MMHTT algorithm by focusing the system's resources on image regions containing the moving objects of interest. The result is a system structure with the potential for automating the establishment of feature point correspondences for the estimation of three-dimensional structure and motion from extended image sequences.

Chapter 6

Summary

The research described in this document has culminated in the development of a new algorithm for the joint detection and tracking of point-source targets in a sequence of digital images. The novelty of this algorithm is the incorporation of a sequential detection algorithm in a 'track-oriented' multiple hypothesis tracking algorithm. The target detection algorithm is used to improve the performance of the tracking algorithm by efficiently combining the target detection and track initiation functions. Conversely, the output of the tracking algorithm is used to improve target detection efficiency by controlling the initiation of new target searches. The resulting algorithm transforms an appropriately processed image sequence into a set of detected target tracks suitable for target state estimation or the estimation of three-dimensional structure and motion.

The analysis of the detection and tracking performance of the MMHTT algorithm, in Chapter 4, suggests that the MMHTT algorithm preserves the detection performance of the MSHT algorithm while significantly extending the length of the detected target trajectories. Although the performance analysis in Chapter 4 is restricted to an evaluation of coarse upper bounds for the detection error probabilities, it significantly extends the performance analysis of the original MSHT algorithm in [40]. The development of a more refined performance analysis is confounded by the magnitude of the branching process for non-trivial candidate trajectory trees and the statistical dependence of image observations from candidate trajectories which share a common node in the candidate trajectory tree.

To recapitulate, the main contributions of this research have been:

1. A new implementation of the Multi-Stage Hypothesis Test algorithm enabling
   (a) a computationally efficient MSHT test initiation procedure, and
   (b) the incorporation of MSHTs in a multiple hypothesis tracking algorithm.
2. The development of a new multiple hypothesis tracking algorithm which
   (a) exploits the MSHT test statistic as a metric for track hypothesis evaluation,
   (b) and employs the MSHT algorithm for target detection and track initiation.
3. An analysis of the performance of the MMHTT and MSHT algorithms including
   (a) the detection error probabilities for trajectories with non-iid observations,
   (b) the detection error probabilities for a candidate trajectory set,
   (c) the detection error probabilities for the MMHTT algorithm,
   (d) performance bounds for the average time to track loss,
4. An application of the MMHTT algorithm to feature detection and tracking.

6.1 Future Work

Several promising avenues exist to extend the current research, including:

A real-time, parallel implementation of the MMHTT algorithm.

A challenging problem would be to develop a parallel implementation of the MMHTT algorithm for real-time implementations. The new implementation of the MSHT algorithm behaves as an array of independent detectors with local communications which may hold promise for an efficient parallel implementation of the target detection and track initiation functions at the image sensor frame rate. The generation and evaluation of track hypotheses could then be implemented on a general purpose processor which polls the local detectors for track confirmations. Several researchers have investigated parallel architectures for weak target detection and tracking in image sequences [43, 38].

A test design procedure for the generalized Gaussian noise distribution.

For the experimental image sequences described in section 5.4, an analysis of the feature maps generated by Debrunner's Gaussian curvature feature generator suggests that the image observations could potentially be better modelled as random variables with a generalized Gaussian probability distribution [64]

$$f_{\mathbf{Y}}(\mathbf{y} \mid \theta) = \frac{k}{2A(k,\sigma^2)\Gamma(1/k)} \exp\left(- \left[\frac{|x|}{A(k,\sigma^2)} \right]^k \right) \tag{6.1}$$

where

$$A(k,\sigma^2) = \left[\sigma^2 \frac{\Gamma(1/k)}{\Gamma(3/k)} \right]^{1/2} \tag{6.2}$$

and

$$\Gamma(k) = \int_0^\infty x^{k-1} e^{-x} \, dx. \tag{6.3}$$

This model has also been successfully employed for target detection and tracking in IR imagery, with the dynamic programming algorithm [36, 34].

As discussed in section 3.1.2, the truncated SPRT, used to evaluate the candidate trajectories in the MMHTT algorithm, is designed as a mixture between a SPRT and a classical FSS test. The SPRT test thresholds are *independent* of the actual probability distribution of the image observations, and depend solely on the desired error probabilities of the test. However, the design expressions for the FSS test threshold τ and the truncation stage K in section 3.1.2 assumed that the image observations were distributed as Gaussian random variables (see (3.18) and (3.19)). A design procedure for K and τ for generalized Gaussian random variables would significantly increase the performance of the MMHTT algorithm for some applications.

Add a model of target dynamics to the evaluation of track hypotheses

The addition of a target dynamics term in the ranking metric for the track hypotheses offers an additional constraint for incorporating *a priori* information to improve the tracking performance of the MMHTT algorithm. The addition of an explicit model of the target dynamics would also allow for a fair comparison of the performance of competing approaches to target detection and tracking such as the dynamic programming algorithm which have successfully exploited a model of target dynamics.

An analysis of the computational requirements of the MMHTT algorithm.

As discussed in Chapter 1, the primary measure of performance for a multitarget detection and tracking algorithm is its computational efficiency in achieving a specified level of detection and tracking performance. Consequently, an ability to estimate the computational requirements of the MMHTT algorithm for arbitrary target scenarios would enable intelligent performance/computation trade-offs to be included in the design. An analysis of the expected number of undecided candidate trajectories would enable an estimate of the algorithm's computational requirements as a function of:

- the memory required to implement the undecided trajectory data structure,
- the memory required to implement each track hypothesis cluster,
- the memory required to store the candidate trajectory set lookup-table,
- and the number of processor operations required for each algorithm function.

.

An extension of the MMHTT algorithm to multi-valued image data.

Multispectral image sequences are the subject of recent research in weak target detection and tracking. It has been shown that the incorporation of spectral information can significantly improve the performance of algorithms designed for weak target detection [25, 71, 39]. An extension of the MMHTT algorithm for multi-valued image data would extend its range of application to include multispectral image sequences.

Symbols and Abbreviations

List of Abbreviations

CCD	charge-coupled device
CFAR	constant false alarm rate
DPA	dynamic programming algorithm
dB	decibels
ECM	electronic countermeasures
EO	electro-optical
FLIR	Forward Looking Infrared
FOV	field-of-view
FSS	fixed sample size
iid	independent and identically distributed
IRST	Infrared Search and Track
JPDA	joint probabilistic data association
MFTDT	Multiframe Target Detection and Tracking
MHT	Multiple Hypothesis Tracking
MMHTT	Multiple Multistage Hypothesis Test Tracking
MSHT	Multi-Stage Hypothesis Test
MTT	multi-target tracking
NN	nearest-neighbour
PDA	probabilistic data association
pdf	probability distribution function
psf	point spread function
RMTI	recursive moving target indicator

ROC	Receiver Operating Characteristic
SB-MHT	Structured Branching Multiple Hypothesis Tracking
SNR	signal-to-noise ratio
SPIE	Society of Photo-Optical Instrumentation Engineers
SPRT	sequential probability ratio test
SSTS	Space-based Surveillance and Tracking Systems

List of Symbols

$A(t)$	time-varying signal amplitude
$\hat{a}$	sequential probability ratio test upper threshold
a	multistage hypothesis test upper threshold
$\mathbf{a}$	vector of multistage hypothesis test upper thresholds
$\hat{b}$	sequential probability ratio test lower threshold
b	multistage hypothesis test lower threshold
$\mathbf{b}$	vector of multistage hypothesis test lower thresholds
$b[x, y, t]$	discrete sample of the imaged background scene radiance
$b[\mathbf{x}, t]$	discrete background image sequence
$\hat{b}[\mathbf{x}, t]$	discrete background image sequence estimate
$\mathbf{b}$	vector of samples of the imaged background scene radiance
C_k	k^{th} track hypothesis cluster
C_{t_k}	log-likelihood ratio of prior probabilities for candidate trajectory t_k
C_{T_k}	log-likelihood ratio of prior probabilities for hypothesized target track T_k
c_0, c_1	truncated sequential probability ratio test weights
D	observation update
$\mathbf{D}_k$	observations in the k^{th} track hypothesis cluster
d	detected target observation
$d()$	photodetector response function
d^2	distance between two signal vectors
G	number of frames in a dynamic programming stage
H_{NT}	new target hypothesis
H_{OT}	old target hypothesis
H_0	target absent hypothesis
H_1	target present hypothesis
H_l	data association hypothesis l
$I(x, y, t)$	time-varying optical image
$I[x, y, t]$	digital image sequence
$I_b(x, y, t)$	time-varying background scene radiance
$I_i(x, y, t)$	time-varying optical image incident on the image sensor

$I_r(x, y, t)$	apparent time-varying scene radiance
K	truncation stage of a truncated sequential probability ratio test
$L()$	likelihood ratio function
L_t	candidate trajectory score function
L_T	track score function
L_H	global tracking hypothesis score function
$N(\mu, \sigma^2)$	Gaussian probability density function with mean μ and variance σ^2
$N[j, k, l]$	discrete noise process
$n[x, y, t]$	discrete sample of the imaging noise process
$\mathbf{n}$	vector of discrete noise samples
$\star\mathbf{n}$	vector of residual noise samples after background clutter suppression
n_{SN}	shot noise process
n_{WB}	wide-band noise process
P_{FA}	probability of false alarm
P_D	probability of detection
$\Pr(x)$	probability of x
(p, s, d)	(position, speed, direction)
$p(x, y)$	optical point spread function
$q()$	sensor quantization function
$\mathcal{R}^3$	three-dimensional space of real numbers
$\mathcal{R}_V^p$	image detector responsivity
r_{max}	maximum radius of target velocity annulus
r_{min}	minimum radius of target velocity annulus
r_o	number of operations required to evaluate $L()$
S	Kalman filter covariance matrix
$S()$	merit function
$\mathbf{s}$	vector of discrete signal samples
$st()$	state transition function
T	target track
T_{opt}	optimal candidate target track
t	candidate target trajectory

ts	dynamic programming target state
$t[x, y, t]$	discrete sample of the imaged target radiance
$\mathbf{t}$	target image
v	target velocity
v_{min}	minimum target velocity
v_{max}	maximum target velocity
Δt	detector integration period (image exposure)
$w_{j,k,l}$	fraction of the total, single exposure target energy
$\Delta x, \Delta y$	detector pixel dimensions
$\mathbf{x}$	discrete image pixel coordinates [x,y]
(x_0, y_0)	target trajectory origin
(x_s, y_s)	detector array sampling point
$(x_{n,j}, y_{n,k})$	relative offset coordinates in tier n of the candidate trajectory set
$(x(t), y(t))$	time-varying target position or target trajectory
Y	random image observation
$\mathbf{Y}$	random vector of image observations
$\mathbf{Y_{t_k}}$	random vector of image observations along the k^{th} candidate trajectory
$\mathbf{Y_{T_k}}$	random vector of image observations along the k^{th} hypothesized target track
$\mathbf{Y_{T_i,k}}$	random vector of image observations along the i^{th} hypothesized target track in the k^{th} cluster
$\mathbf{Y}_\Omega$	random vector of image observations along the candidate trajectories in the undecided trajectory list Ω
y	realization of Y
$\mathbf{y}$	set of received image observations
$f_{\mathbf{Y}}(\mathbf{y} \mid \theta)$	joint pdf of $\mathbf{Y}$ conditioned on the parameter vector θ
$Y \sim f()$	random variable Y has pdf $f()$
α	probability of false alarm
α_{SPRT}	probability of false alarm assigned to the sequential test
α_{FSS}	probability of false alarm assigned to the truncation test
β	probability of missed detection
β_{SPRT}	probability of missed detection assigned to the sequential test

β_{FSS}	probability of missed detection assigned to the truncation test
Γ_0	decision region corresponding with an acceptance of H_0
Γ_1	decision region corresponding with an acceptance of H_1
γ	target occlusion factor
$\delta(\mathbf{y})$	decision rule
$\delta(x, y)$	spatial impulse function
η_{FT}	expected number of false tracks
μ	mean
μ_{I_i}	average image intensity on the sensor focal plane
Ω	the set of candidate trajectories in the current undecided trajectory list
$\Phi()$	standard normalized Gaussian probability distribution function
Φ^{-1}	inverse of Φ
σ^2	variance
τ	decision threshold
τ_{NT}	new target decision threshold
θ	parameter vector of the joint pdf of $\mathbf{Y}$
Θ	the set of all possible values of θ
θ_{H_0}	the set of values of θ conditioned on H_0
θ_{H_1}	the set of values of θ conditioned on H_1

Bibliography

[1] W.N. Martin and J.K. Aggarwal, editors. *Motion Understanding: Robot and Human Vision.* Kluwer Academic Publishers, 1988.

[2] A.N. Venetsanopoulos and V. Cappellini. Real-time image processing. In S.G. Tzafestas, editor, *Multidimensional Systems*, chapter 8, pages 345–394. Marcel Dekker Inc., 1986.

[3] T. S. Huang, editor. *Image Sequence Processing and Dynamic Scene Analysis.* Springer-Verlag, 1983.

[4] T. S. Huang, editor. *Image Sequence Analysis.* Springer-Verlag, 1981.

[5] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications.* Artech House, 1990.

[6] Y. Bar-Shalom and T.E. Fortmann. *Tracking and Data Association.* Academic Press, 1988.

[7] Samuel S. Blackman. *Multiple-Target Tracking with Radar Application.* Artech House, 1986.

[8] S. Mori, C. Chong, E. Tse, and R.P. Wishner. Tracking and classifying multiple targets without a priori identification. *IEEE Transactions on Automatic Control*, AC-31(5):401–408, May 1986.

[9] C.B. Chang and J.A. Tabaczynski. Application of state estimation to target tracking. *IEEE Transactions on Automatic Control*, AC-29:98–109, February 1984.

[10] T.E. Fortmann, Y. Bar-Shalom, and M. Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, OE-8:173–184, July 1983.

[11] D. B. Reid. An algorithm for tracking multiple targets. *IEEE Transactions on Automatic Control*, AC-24(6):843–854, 1979.

[12] Y. Bar-Shalom. Tracking methods in a multitarget environment. *IEEE Transactions on Automatic Control*, 23(4):618–626, 1978.

[13] C.L. Morefield. Application of 0-1 integer programming for multitarget tracking problems. *IEEE Transactions on Automatic Control*, AC-22:302–312, June 1977.

[14] Y. Bar-Shalom and E. Tse. Tracking in a cluttered environment with probabilistic data association. *Automatica*, 11:451–460, September 1975.

[15] P. Smith and G. Buechler. A branching algorithm for discriminating and tracking mulitple objects. *IEEE Transactions on Automatic Control*, AC-20:101–104, February 1975.

[16] J.J. Stein and S.S. Blackman. Generalized correlation of multitarget track data. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11:1207–1217, November 1975.

[17] R.A. Singer, R.G. Sea, and K.B. Housewright. Derivation and evaluation of improved tracking filters for use in dense multitarget environments. *IEEE Transactions on Information Theory*, 20:423–432, July 1974.

[18] R.A. Singer and R.G. Sea. New results in optimizing surveillance system tracking and data correlation performance in dense multitarget environments. *IEEE Transactions on Automatic Control*, AC-18:571–581, December 1973.

[19] R.W. Sittler. An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, MIL-8:125–139, April 1964.

[20] N. Wax. Signal-to-noise improvement and the statistics of track populations. *Journal of Applied Physics*, 26:586–595, May 1955.

[21] H. E. Rauch and O. Firschein. Automatic track assembly for thresholded infrared images. In *Proc. SPIE on Modern Utilization of Infrared Technology VI*, volume 253, pages 75–85, 1980.

[22] N. C. Mohanty. Computer tracking of moving point targets in space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(5):606–611, 1981.

[23] H.E. Rauch, W.I. Futterman, and D.B. Kemmer. Background suppression and tracking with a staring mosaic sensor. *Optical Engineering*, 20(1):103–110, 1981.

[24] A. E. Cowart, W. E. Snyder, and W. H. Ruedger. The detection of unresolved targets using the hough transform. *Computer Vision, Graphics, and Image Processing*, 21:222–238, 1983.

[25] I.S. Reed, R.M. Gagliardi, and H.M. Shao. Application of three-dimensional filtering to moving target detection. *IEEE Transactions on Aerospace and Electronic Systems*, 19(6):898–905, 1983.

[26] Y. Barniv. Dynamic programming solution for detecting dim moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(1):144–156, 1985.

[27] L. T. Bruton and N. R. Bartley. The enhancement and tracking of moving objects in digital images using adaptive three-dimensional recursive filters. *IEEE Transactions on Circuits and Systems*, CAS-33(6):604–612, 1986.

[28] Y. Barniv and O. Kella. Dynamic programming solution for detecting dim moving targets. part ii: Analysis. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(6):776–788, 1987.

[29] P. L. Chu. Optimal projection for multidimensional signal detection. *IEEE Transactions on ASSP*, 36(5):775–786, 1988.

[30] I. Reed, R. Gagliardi, and L. Stotts. Optical moving target detection with 3-d matched filtering. *IEEE Transactions on Aerospace and Electronic Systems*, 24(4):327–335, 1988.

[31] Y. Chen. On suboptimal detection of 3-dimensional moving targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-25(3):343–350, 1989.

[32] W.B. Kendall, A.D. Stocker, and W.J. Jacobi. Velocity filter algorithms for improved target detection and tracking with multiple-scan data. *Signal and Data Processing of Small Targets 1989*, Proc. SPIE 1096:127–137, 1989.

[33] S.C. Pohlig. An algorithm for detection of moving optical targets. *IEEE Transactions on Aerospace and Electronic Systems*, AES-25(1):56–63, 1989.

[34] J. Arnold and H. Pasternack. Detection and tracking of low-observable targets through dynamic programming. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:207–217, 1990.

[35] G.C. Demos, R.A. Ribas, T.J. Broida, and S.S. Blackman. Applications of mht to dim moving targets. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:53–62, 1990.

[36] M.F. Fernandez, A. Aridgides, and D. Bray. Detecting and tracking low-observable targets using ir. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:193–206, 1990.

[37] B. Porat and B. Friedlander. A frequency domain algorithm to mutiframe detection and estimation of dim targets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(4):398–401, 1990.

[38] K. Preston Jr. Detection of weak, subpixel targets using mesh-connected cellular automata. *IEEE Transactions on Aerospace and Electronic Systems*, 26(3):548–558, 1990.

[39] I. Reed, R. Gagliardi, and L. Stotts. A recursive moving-target-indication algorithm for optical image sequences. *IEEE Transactions on Aerospace and Electronic Systems*, 26(3):434–439, 1990.

[40] S.D. Blostein and T.S. Huang. Detecting small, moving objects in image sequences using sequential hypothesis testing. *IEEE Trans. Sig. Proc.*, 39(7):1611–1629, 1991.

[41] D. Braunreiter and N. Banh. Detection of moving subpixel targets in infrared clutter with space-time filtering. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:73–83, 1991.

[42] N. Carender and D. Casasent. Point target detection, location and track initiation: Initial optical lab results. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:35–48, 1991.

[43] A.D. Stocker and P. Jensen. Algorithms and architectures for implementing large velocity filter banks. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:140–155, 1991.

[44] Z. Mao and R.N. Strickland. Image sequence processing for target estimation in forward-looking-infrared imagery. *Optical Engineering*, 27(7):541–549, 1988.

[45] B. Bhanu. Automatic target recognition: State of the art survey. *IEEE Transactions on Aerospace and Electronic Systems*, AES-22(4):364–379, 1986.

[46] A.L. Gilbert, M.K. Giles, G.M. Flachs, R.B. Rogers, and Y.H. U. A real-time video tracking system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:47–56, 1980.

[47] Oliver E. Drummond, editor. *Signal and Data Processing of Small Targets 1989*, volume SPIE Vol. 1096. The Society of Photo-Optical Instrumentation Engineers, SPIE, March 1989. Orlando, Florida.

[48] Oliver E. Drummond, editor. *Signal and Data Processing of Small Targets 1990*, volume SPIE Vol. 1305. The Society of Photo-Optical Instrumentation Engineers, SPIE, April 1990. Orlando, Florida.

[49] Oliver E. Drummond, editor. *Signal and Data Processing of Small Targets 1991*, volume SPIE Vol. 1481. The Society of Photo-Optical Instrumentation Engineers, SPIE, April 1991. Orlando, Florida.

[50] T.S. Axelrod, N.J. Colella, and A.G. Ledebuhr. The wide-field-of-view camera. *Energy and Technology Review*, December 1988.

[51] S. D. Blostein. Multidimensional signal detection for wide field of view telescope applications. Technical report, Dept. of Electrical Engineering Research Report 90-2, Queen's University, 1990.

[52] J.F. Bronskill, J.S.A. Hepburn, and W.K. Au. Detection and classification of target formations in infrared image sequences. In M.J. Triplett A.J. Huber and J.R. Wolverton, editors, *Imaging Infrared: Scene Simulation, Modeling and Real Image Tracking*, pages 150–164, Orlando, Florida, March 1989. The Society of Photo-Optical Instrumentation Engineers, The Society of Photo-Optical Instrumentation Engineers.

[53] David S.K. Chan, D.A. Langan, and D.A. Staver. Spatial processing techniques for the detection of small targets in ir clutter. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:53–62, 1990.

[54] G. Wang, R.M. Inigo, and E.S. McVey. A pipeline algorithm for detection and tracking of pixel-sized target trajectories. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:53–62, 1990.

[55] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images - a review. *Proc. IEEE*, 76(8):917–935, 1988.

[56] T.J. Broida, S. Chandrashekhar, and R. Chellappa. Recursive 3-d motion estimation from a monocular image sequence. *IEEE Trans. on AES.*, 26(4):639–656, 1990.

[57] C. Jerian and R. Jain. Polynomial methods for structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(12):1150–1165, 1990.

[58] J. Weng, T.S. Huang, and N. Ahuja. 3-d motion estimation, understanding and prediction from noisy image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):370–389, 1987.

[59] Y.-L. Chang and J.K. Aggarwal. 3d structure reconstruction from an ego motion sequence using statistical estimation and detection theory. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 268–273, 1991.

[60] C.H. Debrunner. *Structure and Motion from Long Image Sequences.* PhD thesis, University of Illinois at Urbana-Champaign, 1990.

[61] Joseph W. Goodman. *Statistical Optics.* John Wiley and Sons, 1985.

[62] Eustace L. Dereniak and Devon G. Crowe. *Optical Radiation Detectors.* John Wiley and Sons, 1984.

[63] H. V. Poor. *An Introduction to Signal Detection and Estimation.* Springer-Verlag, 1988.

[64] Saleem A. Kassam. *Signal Detection in Non-Gaussian Noise.* Springer-Verlag, 1988.

[65] H. L. Van Trees. *Detection, Estimation, and Modulation Theory, Part I.* John Wiley and Sons, 1968.

[66] C.D. Wang. Adaptive spatial/temporal/spectral filters for background clutter suppression and target detection. *Optical Engineering*, 21(6):1033–1038, 1982.

[67] T.J. Patterson, D.M. Chabries, and R.W. Christiansen. Detection algorithms for image sequence analysis. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 37(9):1454–1458, 1989.

[68] M. Fernandez, A. Aridgides, D. Randolph, and D. Ferris. Optimal subpixel-level frame-to-frame registration. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:172–179, 1991.

[69] A.S. Elfishawy, S.B. Kesler, and A.S. Abutaleb. Adaptive algorithms for change detection in image sequences. *Signal Processing*, 23(2):179–191, May 1991.

[70] T. Soni, B.D. Rao, J.R. Zeidler, and W.H. Ku. Enhancement of images using the 2-d lms adaptive algorithm. In *Proceedings of the 1991 International Conference*

on Acoustics, Speech and Signal Processing, pages 3029–3032, Toronto, May 1991. IEEE.

[71] A. Margalit, I. S. Reed, and R. M. Gagliardi. Adaptive optical target detection using correlated images. *IEEE Transactions on Aerospace and Electronic Systems*, AES-21(3):394–405, 1985.

[72] J.Y. Chen and I.S. Reed. A detection algorithm for optical targets in clutter. *IEEE Transactions on Aerospace and Electronic Systems*, AES-23(1):46–59, 1987.

[73] A. Aridgides, M. Fernandez, D. Randolph, and D. Bray. Adaptive three-dimensional spatio-temporal filtering techniques for infrared clutter suppression. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:63–74, 1990.

[74] A.D. Stocker, I.S. Reed, and X. Yu. Multi-dimensional signal processing for electro-optical target detection. *Signal and Data Processing of Small Targets 1990*, Proc. SPIE 1305:218–219, 1990.

[75] A. Aridgides, M. Fernandez, D. Randolph, and D. Ferris. Adaptive 4-d ir clutter suppression filtering techniques. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:110–116, 1991.

[76] D.S.K. Chan. Model-based analysis of 3d spatial-temporal ir clutter suppression filtering. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:117–128, 1991.

[77] M.R. Wohlers. Bounds on the performance of optimal four dimensional filters for detection of low contrast ir point targets. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:129–139, 1991.

[78] L.B. Milstein and T. Lazicky. Statistical tests for image tracking. *Computer Graphics and Image Processing*, 7:413–424, 1978.

[79] Y.Z. Hsu, H.-H. Nagel, and G. Rekers. New likelihood test methods for change detection in image sequences. *Computer Vision, Graphics, and Image Processing*, 26:73–106, 1984.

[80] I.S. Reed and X. Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 38(10):1760–1770, 1990.

[81] R.E. Warren. Generalized likelihood ratio detector for small targets in stationary background clutter. *Signal and Data Processing of Small Targets 1990*, pages 24–40, April 1990.

[82] A. Wald and J. Wolfowitz. Optimum character of the sequential probability ratio test. *Ann. Math. Stat.*, 19(3):326–339, 1948.

[83] S. Tantaratana and H. V. Poor. Asymptotic efficiencies of truncated sequential tests. *IEEE Transactions on Information Theory*, IT-28(6):911–923, 1982.

[84] R. Bellman. *Dynamic Programming.* Princeton University Press, 1957.

[85] Y. Bar-Shalom, editor. *Multitarget-Multisensor Tracking: Advanced Applications*, chapter Dynamic Programming Algorithm for Detecting Dim Moving Targets. Artech House, 1990.

[86] A.B. Poore and N. Rijavec. Multi-target tracking and multi-dimensional assignment problems. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:345–356, 1991.

[87] J.S. Bird and M.M. Goulding. Hit-rate constrained target detection. In *International Symposium on Information Theory and its Applications '90*, pages 295–297, 1990.

[88] A. Aho, J.E. Hopcroft, and J.D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.

[89] B.E. Fridling and O.E. Drummond. Performance evaluation methods for multiple target tracking algorithms. *Signal and Data Processing of Small Targets 1991*, Proc. SPIE 1481:371–383, 1991.

[90] O.E. Drummond and S.S. Blackman. Challenges of developing algorithms for multiple sensor, multiple target tracking. *Signal and Data Processing of Small Targets 1989*, Proc. SPIE 1096:244–255, 1989.

[91] L. Aroian and D. Robison. Direct methods for exact truncated sequential tests of the mean of a normal distribution. *Technometrics*, 11(4):661–675, 1969.

[92] Richard J. Larsen and Morris L. Marx. *An Introduction to Probability and Its Applications.* Prentice-Hall, 1985.

[93] C. Moler D. Kahaner and S. Nash. *Numerical Methods and Software.* Prentice Hall, 1989.

[94] J.W. Roach and J.K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:554–562, 1980.

[95] Y. Liu and T.S. Huang. A linear algorithm for motion estimation using straight line correspondences. *Computer Vision, Graphics and Image Processing*, 44:35–57, 1988.

[96] T.S. Huang and C.H. Lee. Motion and structure from orthographic projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11:536–540, May 1989.

[97] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from two perspective views: algorithms, error analysis and error estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(5):451–476, 1989.

[98] V. Salari and I.K. Sethi. Feature point correspondence in the presence of occlusion. *IEEE Trans. PAMI*, 12(1):87–91, 1990.

[99] V. Venkateswar and R. Chellappa. Hierarchical feature based matching for motion correspondence. In *Proc. IEEE Workshop on Visual Motion*, 1991.

[100] G.-Q. Wei, Z. He, and S.D. Ma. Fusing the matching and motion estimation of rigid point patterns. In *1990 IEEE International Conference on Robotics and Automation*, pages 2017–2022, 1990.

[101] S.T. Barnard and W.B. Thompson. Disparity analysis of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2:333–340, 1980.

[102] S. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12:269–275, 1980.

[103] J.-Q. Fang and T.S. Huang. Some experiments on estimating the 3-d motion parameters of a rigid body from two consecutive image frames. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(5):481–490, September 1984.

[104] I.K. Sethi and R. Jain. Finding trajectories of feature points in a monocular image sequence. *IEEE Trans. PAMI*, 9(1):56–73, 1987.

[105] V.S.S. Hwang. Tracking feature points in time-varying images using an opportunistic selection approach. *Pattern Recognition*, 22(3):247–256, 1989.

[106] G. Verghese, K.L. Gale, and C.R. Dyer. Real-time motion tracking of three-dimensional objects. In *1990 IEEE International Conference on Robotics and Automation*, pages 1998–2003, 1990.

[107] H. Harlyn Baker. Motion tracking on the spatiotemporal surface. In *Workshop on Visual Motion, October 1991*, 1991. Do not reference this until after the workshop and check the ref.

[108] C. Debrunner and N. Ahuja. Motion and structure factorization and segmentation of long image sequences. In *Proc. of the 2'nd ECCV*, Ligure, Italy, May 1992.

[109] Michael E. Mortenson. *Geometric Modeling.* John Wiley and Sons, 1985.

[110] Jerrold E. Marsden and Anthony J. Tromba. *Vector Calculus.* W.H. Freeman and Company, 1981.

[111] K.P. Karmann, A.v.Brandt, and R. Gerl. Moving object segmentation based on adaptive reference images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1990. Submitted 1990.

[112] G.W. Donohoe, D.R. Hush, and N. Ahmed. Change detection for target detection and classification in video sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing 1988*, pages 1084–1087, 1988.

[113] K.P. Karmann and A.V. Brandt. Detection and tracking of moving objects by adaptive background extraction. In *Proceedings of the 6'th SCIA,Oulu,Finland*, 1989.

[114] G. Ward, F. Rubinstein, and R. Clear. A ray tracing solution for diffuse interreflection. *Computer Graphics*, 22(4), 1988.

[115] G.-S.J. Young and R. Chellappa. 3-d motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):735–759, 1990.

Vita

Haydn Scott Panther Richardson

Born: Scarborough, Ontario, 1967.

EDUCATION

Queen's University	M.Sc.	Electrical Engineering	1990–92
Queen's University	B.A.Sc.	Mathematics and Engineering	1985–89
Queen's University	School of Graduate Studies Dean's Award		1991
Natural Sciences and Engineering Research Council Post-Graduate Scholarship			1992

PUBLICATIONS AND PRESENTATIONS

Haydn Richardson and Steven Blostein, (1992) "A Sequential Detection Framework for Feature Tracking within Computational Constraints", *1992 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.*

Haydn Richardson and Steven Blostein, (1991), "Image Sequence Processing for Motion Detection and Estimation in Unstructured Environments", *Proceedings of the Seventh IEEE Workshop on Multidimensional Signal Processing,* 5.4.

S.D. Blostein and H.S. Richardson, (1990)," Multidimensional Autoregressive Parameter Estimation Using Iteratively Reweighted Least Squares", *Proceedings of the 1990 IEEE International Conference on Acoustics, Speech and Signal Processing,* 2699:2702.