Context-Aided Tracking with Adaptive Hyperspectral Imagery

THESIS

Andrew C. Rice, Civilian

AFIT/GE/ENG/11-43

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/11-43

Context-Aided Tracking with Adaptive Hyperspectral Imagery

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Andrew C. Rice, B.S.C.S.

Civilian

June 2011

AFIT/GE/ENG/GE-11J

Context Aided Tracking with Adaptive Hyperspectral Imagery

Andrew C. Rice, B.S.C.S.
Civilian

Approved:

Dr. Juan R. Vasquez (Chair)	24-May-2011 date
Maj Michael J. Mendenhall, Ph.D. (Member)	24-MAY-2011 date
Dr. John F. Raquet (Member)	24 MAY 2011 date

AFIT/GE/ENG/11-43

Abstract

A methodology for the context-aided tracking of ground vehicles in remote airborne imagery is developed in which a background model is inferred from hyperspectral imagery. The materials comprising the background of a scene are remotely identified and lead to this model. Two model formation processes are developed: a manual method and a novel autonomous method that exploits an emerging adaptive, multiple-object-spectrometer instrument. A multiple-hypothesis-tracker is incorporated, which utilizes background statistics to form track costs and associated track maintenance thresholds. Traditionally, these statistics are uniform constants, but the advent of the background model allows for spatially-varying statistics. In an experiment designed to isolate the problem into a simple and parametric single-target situation, context-aided tracking is shown to improve aggregate tracking performance by 50% in certain operating conditions. A semi-automated background modeling approach is shown to qualitatively arrive at a reasonable background model with minimal operator intervention. A novel, adaptive, and autonomous approach is given which converges to a 66% correct background model in $\frac{1}{18}^{\text{th}}$ the time of the baseline – a 95% reduction in sensor acquisition time – then transitions to the 100% correct model in the steady-state. Finally, the context-aided system is demonstrated in a high-fidelity tracking testbed. The context-aided tracking improves certain aggregate tracking metrics by 4% relative to a system using uniform background statistics – an important finding nevertheless diluted by benign content in the scenario. Another metric is proposed as the most salient measure of performance gain in the context-aided system, showing a dramatic *30% reduction in error* relative to the best-performing uniform background statistic system. The analysis demonstrates that context-aided tracking with adaptive hyperspectral data is a viable approach.

Acknowledgements

I would like to express my gratitude to all those who have enthusiastically supported my studies and made this document possible.

First, to my family, whom have endured my excessively lengthy graduate study with grace and support. I thank you for granting me time and patience for this pursuit, and apologize for the many times I have been wanted and found indisposed. To my wife, I thank you for your unconditional commitment and support. To my children, I thank you for your patience, which you've each shown in portions beyond your years. To my parents and parents-in-law, thank you for your encouragement, prayers for academic expediency, and especially for picking up my domestic slack: children watched, cars repaired, and lawns mowed.

Next, to my advisor Dr. Juan Vasquez. Upon our initial meeting, I was left with two impressions: that I did not know nearly as much about tracking as I thought I did, and that I had just met an unsurpassed mentor from whom to learn. You have taught me a great deal about matters theoretical and practical, and it has been an honor to work with you. The convergence of a firm command of a field of study with the ability to teach it lucidly and patiently is a rare gift. You are truly a gentleman and a scholar.

I sincerely appreciate the support and academic excellence of the members of my committee: Dr. Vasquez, Maj. Mendenhall, and Dr. Raquet. The sponsorship of AFRL/RYA in this research endeavor is appreciated, in particular the interest and hyperspectral expertise of my colleague and AFIT alumna Karmon Vongsy, who was instrumental in collecting a portion of the hyperspectral data used in this study four years ago – which now seems like an eternity.

And to my colleagues and friends, you have been a pleasure to work with. From the first AFIT Tecˆ Edge crew of four years ago who planted the seeds for context-aided tracking, to my colleagues at AFRL and industry who provided the tools, data, and hyperspectral know-how, I owe many debts of gratitude. In particular, AFIT

alumnus Neil Paris [39] pioneered the semi-automated background modeling work which was critical to this research. To all of my office-mates over the past three years – Juan, Chris, Sarah, Karl, Neil, Leo, Daniel, and Ryan – thank you for making work a fun place to be. I will always cherish your friendship. We've exploited hyperspectral like it's going out of style (which hopefully it is not), hunting everything from tanks to tortoises. We've done small-UAV and counter-small-UAV (some irony, that). And, we discovered a way to play video games for science, which is surely our crowning achievement. As we go our separate ways to AFRL, industry, academia, the clergy, and wherever our dreams and skills take us, I wish each of you success and happiness.

Andrew C. Rice

Table of Contents

Page

List of Figures

Figure Page

List of Tables

List of Abbreviations

Abbreviation		Page
GRLVQ	Generalized Relevance Learning Vector Quantization	25
SOM	Self Organizing Map	25
ATR	Automated Target Recognition	27
VIVID	Video Verification of Identity	27
AGRLVQI	Adaptive Generalized Relevance Learning Vector Quantization Improved	35
OC's	Operating Conditions	63
SUMO	Simulation of Urban Mobility	66
ELC	Emperical Line Calibration	78
USGS	United States Geological Survey	78
NEFDS	Nonconventional Exploitation Factors Dataset	78
ASTER	Advanced Spaceborne Thermal Emission Reflection Radiometer	78
MEX	`Matlab`® Extensions	91
DTED	Digital Terrain Elevation Data	92
ENU	Easting, Northing, Up	92

Context-Aided Tracking with Adaptive Hyperspectral Imagery

I. Introduction

The requirement to remotely track ground vehicles within urban environments is becoming increasingly pervasive in both civilian and military contexts. These environments are typified by high vehicle density, agility, and diversity, as well as frequent occlusions. There exists a stark contrast between tracking missions conceived by the United States Air Force (USAF) several decades ago and those encountered operationally today. Many currently operational USAF systems were designed to detect and track foreign military ground order of battle (GOB) targets. While camouflage, concealment, and deception (CC&D) techniques can make these targets difficult to detect, they frequently exist in isolated, rural environments. They are also explicitly military in appearance and employment. Today's missions, however, often consist of warfare against an asymmetric force – a concept which became more pervasive during the last decade's global war on terrorism (GWOT). The threat, then, may be embedded within a crowded urban environment and assume the appearance and behavior of daily civilian life. In this context, the ability to prosecute a threat is best enabled by vigilantly maintaining awareness of all entities within the area of regard. Indeed, the stated USAF science and technology (S&T) vision is, "Anticipate, find, fix, track, target, engage, and assess anything, anywhere, anytime" (AF2T2EA4) [8]. In this concept, every available intelligence cue must be "force multiplied" by tracking systems, which are increasingly called upon to be persistent, pervasive, and highly accurate with respect to track identities. Simply put, the tracking system must maintain a strict one-to-one mapping between objects in the area of regard, and unique track identities within the system. In order to realize the AF2T2EA4 mandate, the Air Force Research Laboratory (AFRL) has developed eight focused long term challenges

(FLTC), two of which are directly motivated by the urban tracking challenge [48] (emphasis added):

> FLTC#2, Unprecedented Proactive Surveillance and Reconnaissance, is focused on the ability to **continuously** detect, **track** and ID critical threats to anticipate and deliver effects **anywhere**, including within an anti-access environment.

> FLTC #3, Dominant Difficult **Surface Target** Engagement/Defeat, is focused on the ability to deliver selectable and scaleable non-lethal or lethal effects against adversaries and/or their support activities, improvised explosive devices (IEDs), and chemical, biological, radiological, and nuclear explosive (CBRNE) threats **in an urban warfare environment.**

Finally, in the Tier-1 Strategic S&T Goals for AFRL, the highest priority 2015 Strategic Goal for multi-layer sensing architectures is stated as [8] (emphasis added):

> Demonstrate a layered and flexible sensing architecture that responds to the commanders intent by anticipating, detecting, **continuously tracking,** identifying, and precisely locating – **with high confidence** – greater than 80% of selected high-value difficult targets (e.g., **urban,** low-observable cruise missiles, buried); initially to be demonstrated in the air domain.

1.1 Problem Statement and Key Challenges

While the USAF is well poised to meet the strategic goals of the S&T vision into 2015 and beyond, there remains a subtlety of AF2T2EA4 which will remain challenging for the foreseeable future: mission-relevant tracker performance. Concisely, the problem statement of this thesis is to quantify the negative impact of background elements on tracker effectiveness, and to lessen that impact through the use of context.

Much of what is known of tracking system capabilities and performance arise from the earliest – and relatively constrained – tracking problems such as air defense, and air-to-air engagement systems. These problem domains began to test the capability of tracking algorithms to handle several targets in the presence of clutter. Within these paradigms, the key tracking performance metrics have generally dealt with how accurately the tracker estimates the state of a moving object. However,

the notion of tracking "anything, anytime, anywhere", e.g., difficult surface targets in the urban environment, "with high confidence" calls for a redefinition of tracker performance. In this problem domain, high kinematic – i.e., position and velocity – accuracy is merely a prerequisite rather than the end goal. The more challenging problem is to maintain tracks with pure identity through ambiguity, e.g., merging targets, and severe occlusion. This can be observed as a chain of custody, in which the identity of each object must be consistent for as long as it is tracked. The failure to preserve a single, consistent identity for an object has negative consequences in forensic applications, e.g., tracking a hostile vehicle backwards in time from the scene of an attack to its origin. Similarly, rules of engagement (ROE) enforce a strict chain of custody on a target as it is tracked by an ISR system through handoff to a strike platform for prosecution. Even a single break in the chain of custody results in mandatory reestablishment of the engagement – and possibly a failed strike. There exist challenges that do little to degrade kinematic accuracy, but play havoc with the purity of track identities:

- Track identity loss occurs when the observations of an object no longer support the tracker's ability to maintain the track, and it is deleted. Subsequently, the measurements resume and a new track with a new identity is formed. Here, the chain of custody is broken for that object, and the tracker's estimate of the number of objects will be inflated. Certainly there are cases in which this is unavoidable, e.g., a vehicle disappears within a parking garage for an extended period of time. Reestablishing the proper identity to the reemerging vehicle in such a challenging case is perhaps possible with feature-aided tracking (FAT) and track stitching techniques. However, there exists a broad family of cases – e.g., a vehicle travelling underneath a moderately long and dense tree canopy causing frequent measurement loss – in which the context supports less aggressive track deletion in hopes of coasting the track through unscathed.
- Track swaps occur when closely spaced objects meet and exchange track identities. Here, the chain of custody is polluted for both tracks.

- Track divergence occurs when the observations of an object no longer support tracking, yet the tracker does not delete the track quickly enough. Instead, the track coasts until it encounters the measurements of a different object, which it locks onto. Here, the chain of custody is polluted for the track, and the tracker's estimate of the number of objects will be understated.

There exists, then, a strong motivation for providing the tracker with as much information as possible with which to make sound track maintenance decisions. All other things being equal, a tracker that knows under which circumstances to delete a track quickly, and when to aggressively coast a track, is likely to have better track identity performance than a tracker without such information. While there are certainly many ways to gain this information and then to apply it to the tracking logic, an especially attractive approach is to use a mechanism which already exists in the tracker and has some analytical meaning. The tracking systems of interest in this study are probabilistic in nature and therefore form statistical models of all elements of the system; this includes the background itself, as well as the manner in which it affects the measurement process. Using the term "context" to refer to these properties of the background, the application of these properties is context-aided tracking (CAT). Primarily, CAT seeks to improve tracker performance by emphasizing track identity purity. The types of background statistics relevant to CAT have much to do with the observation mechanism – i.e., the sensor and signal processor – and are difficult to measure directly. One approach is to learn these statistics over time from the tracking system itself. This learning process, however, is partially hampered without the corresponding truth of the targets themselves. A more abstract approach to forming background statistics is to determine a mapping from classes of background objects, e.g., trees, roads, buildings, into the functional impact those objects have on the target tracking process. In this approach, it is sufficient to know the placement of background objects in the scene and then to apply the mapping to arrive at the CAT statistics. Fortunately, remote hyperspectral sensing is well suited to the challenge of determining the location of background elements.

The track identity problem is further compounded by the complexity of the urban tracking challenge. While it would be satisfying to assume that tracker computational complexity is a linear function of the number of targets, $O(n)$ in computer science parlance – that is unfortunately not the case. In fact, a multiple-target tracking system must manage the combinatorics between tracks and measurements at each frame, and possibly do so over a window of history frames. The upper-bound on complexity is $O(n^2)$, and perhaps far worse. The foundation for the solution to this problem has fortunately already been laid in the form of the multiple hypothesis tracker.

1.2 Current System Capabilities

1.2.1 Hyperspectral Imaging Sensors. Hyperspectral imaging (HSI) sensors are a relatively new field of remote sensing technology. The phenomenology of hyperspectral sensing is discussed in Section 2.1. Applications for HSI sensors fall into distinct civil and military categories. Classical military applications of HSI technology lie within measurement and signature intelligence (MASINT). These employments tend to utilize expensive HSI sensors on a small population of highly classified intelligence, surveillance, and reconnaissance (ISR) vehicles. The tasking and exploitation of MASINT sensors generally occurs offline via dedicated ISR organizations which are not a time-critical part of the AF2T2EA4 loop. The National Air and Space Intelligence Center (NASIC) is an example of a military organization concerned with MASINT. These take advantage of HSI sensor capabilities to detect material and chemical signatures which manifest subtly in reflected visible or infrared light. Counter camouflage, concealment and deception CCC&D techniques use HSI to detect targets which were intentionally hidden from sight. CBRNE detection techniques use HSI to detect the evidence of weapons of mass destruction (WMD) manufacture or deployment, e.g., plumes or runoff. As the first tactical HSI sensors become available, emerging military uses include target detection via *a priori* signature matching, and improvised explosive device (IED) detection, which is concerned with detecting the

evidence of IED emplacement. Sensors which the USAF currently uses for research or as operational systems include:

- Airborne Cueing and Exploitation System Hyperspectral (ACES HY), providing HSI capabilities for the MQ-1 Predator Unmanned Aircraft System (UAS) and other manned and unmanned aircraft. FY09-FY15 estimate $35M [5].
- The AFRL Spectral Infrared Remote Imaging Transition Testbed (SPIRITT). Prior years $45M, $16M FY10-FY11 [5].
- The Hyperspectral Collection and Analysis System (HYCAS) advanced concept technology demonstration (ACTD), which employs an HSI instrument and the capability of "tasking, processing, exploitation, and dissemination, the TPED side of really operationalizing and institutionalizing the use of hyperspectral" [1].
- The AFRL Advanced Responsive Tactically Effective Military Imaging Spectrometer (ARTEMIS), which is a tactically-taskable HSI instrument aboard the TacSat-3 satellite [38].

Civil uses for HSI sensors have traditionally been closely aligned with academic earth science studies. The exemplary civil HSI instrument is the National Aeronautics and Space Administration (NASA) AVIRIS (Airborne Visible/Infrared Imaging Spectrometer), which images the spectral region from 0.4μ to 2.5μ at 0.01μ resolution. While AVIRIS played a role as an airborne testbed for instruments with military purposes [51], it was also designed to advance environmental, geological, and pollution studies [21]. Indeed, the notion of using HSI imagery to derive thematic maps, i.e., the material composition of broad areas of land, is foundational to this study in context-aided-tracking. AVIRIS products have also historically been important in the study of potential markers for climate change [58]. Many other civil airborne HSI sensors are available for hire today, and serve in academic, agricultural – such as precision farming [25], oil exploration [19], and oil spill mitigation [49].

1.2.2 Persistent Surveillance. A significant paradigm shift in military doctrine is taking place, wherein certain "reconnaissance" missions are being supplanted by "persistence" missions [40]. Persistent surveillance implies continuous, multimodal, integrated data collection with low-latency dissemination and on-demand exploitation. A variety of sensors in the USAF inventory are meeting this need, and new research efforts are set to bolster the capability:

- The Joint Surveillance Target Attack Radar System (Joint STARS) is an airborne radar system that provides persistent ground surveillance of a corps-sized region. [7]
- Gorgon Star provides city-sized broad area surveillance capability for Combatant Commanders, $115M FY09-FY11 [5].
- Wide Area Airborne Surveillance (WAAS) Program of Record, providing flexible, end-to-end, persistent surveillance of city-sized areas on various manned and unmanned aircraft, $253M FY11-FY15 [5].
- The Autonomous Real-time Ground Ubiquitous Surveillance-Imaging System (ARGUS-IS) is a DARPA-funded, gigapixel-class color video imager with framerates up to 12Hz, providing unprecedented persistent surveillance capabilities aboard an A-160 Hummingbird UAS [34]. The Autonomous Real-time Ground Ubiquitous Surveillance - Infrared (ARGUS-IR) enhancement adds wide-area night surveillance capabilities. Combined $58M FY10-FY12 [6].

Persistent surveillance provides the crucial underpinning moving intelligence (MOVINT) required for ground target tracking. As such, it represents another foundational element to this study in CAT.

1.2.3 GIS. Geographic information systems (GIS) are databases which contain information tied to a terrestrial spatial coordinate system. There are many civil uses for GIS, including municipal planning, utilities management, and disaster management. Military uses for GIS include mission planning, logistics, targeting, and to

support intelligence collection and processing. The Department of Defense (DOD) has largely allocated geographic intelligence (GEOINT) creation duties to the National Geospatial-Intelligence Agency (NGA). GEOINT is derived from many types of sensors and platforms operating in many different modalities, including airborne HSI. Once in a GIS database, this data has generally been heavily processed to reduce its size or to abstract the signal into a more functional representation. With respect to CAT, GIS data is the first-order source for context information, to include road networks, target hospitability, and land-cover. Since GIS products are generally produced according to a reconnaissance and processing schedule, they may lack contemporary changes and diurnal environmental effects. The notion of extracting contemporary GIS from a theater-level HSI asset is relatively new, and has been motivated by the increasing availability of such sensors. This study on CAT will further explore the utility of GIS data derived from contemporary HSI in a semi-automated or automated fashion.

1.3 Contributions

Here, two promising areas of research are combined to cope with the challenges of tracking difficult ground targets in an urban context: context-aided tracking and adaptive hyperspectral sensing. The availability of novel adaptive hyperspectral sensors has led to new sensor resource management methods providing spatially varying background statistics of the surveillance region. These statistics are then incorporated into a multiple hypothesis tracking system to enable more robust tracking in the presence of vehicle occlusion typically encountered in urban environments. An overview of the system architecture is given in Figure 1.1; additional detail of the background modeling architecture is given in Figure 1.2. These architecture Figures will be referenced throughout this document to aid the reader in understanding the development of the primary functions.

In an experiment designed to isolate the problem into a simple and parametric single-target situation, context-aided tracking will be clearly shown to *improve aggre-*

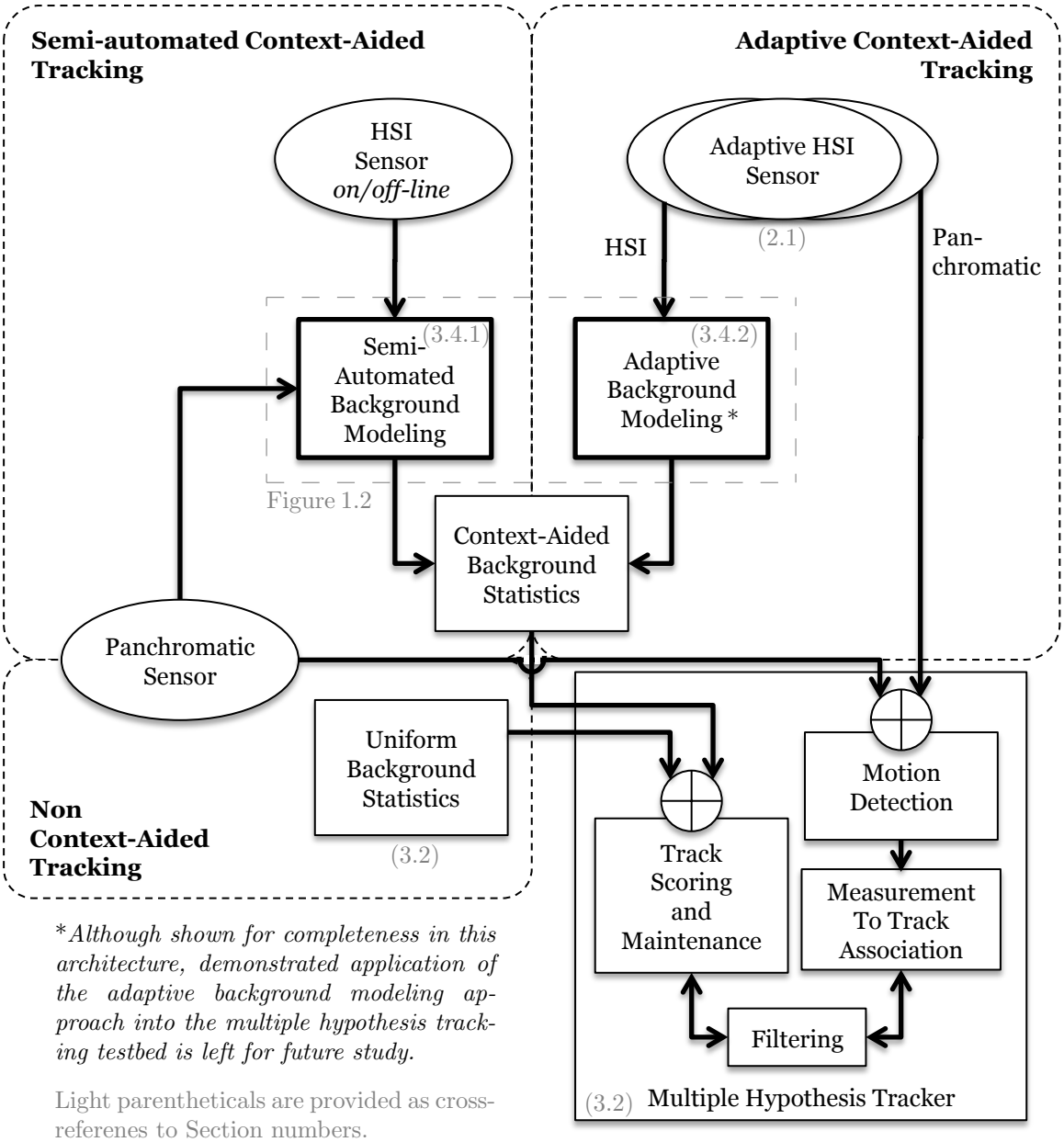


Figure 1.1: An illustration of the tracking architecture in this study. The key differentiators of the systems are the derivation of background statistics and the requisite sensors. Clockwise from the lower left: The non-context-aided tracking system utilizes uniform background statistics and requires a panchromatic sensor. This represents the baseline capability. The semi-automated context-aided tracking system utilizes context-aided statistics. It requires a panchromatic sensor as well as a hyperspectral sensor, which may generate online (mission) or offline (pre-mission) data. The adaptive context-aided tracking system also uses context-aided statistics. It requires an adaptive hyperspectral sensor, providing both hyperspectral and panchromatic data. All tracking systems incorporate a multiple hypothesis tracker, which performs a core set of tracking, association, and maintenance functions.

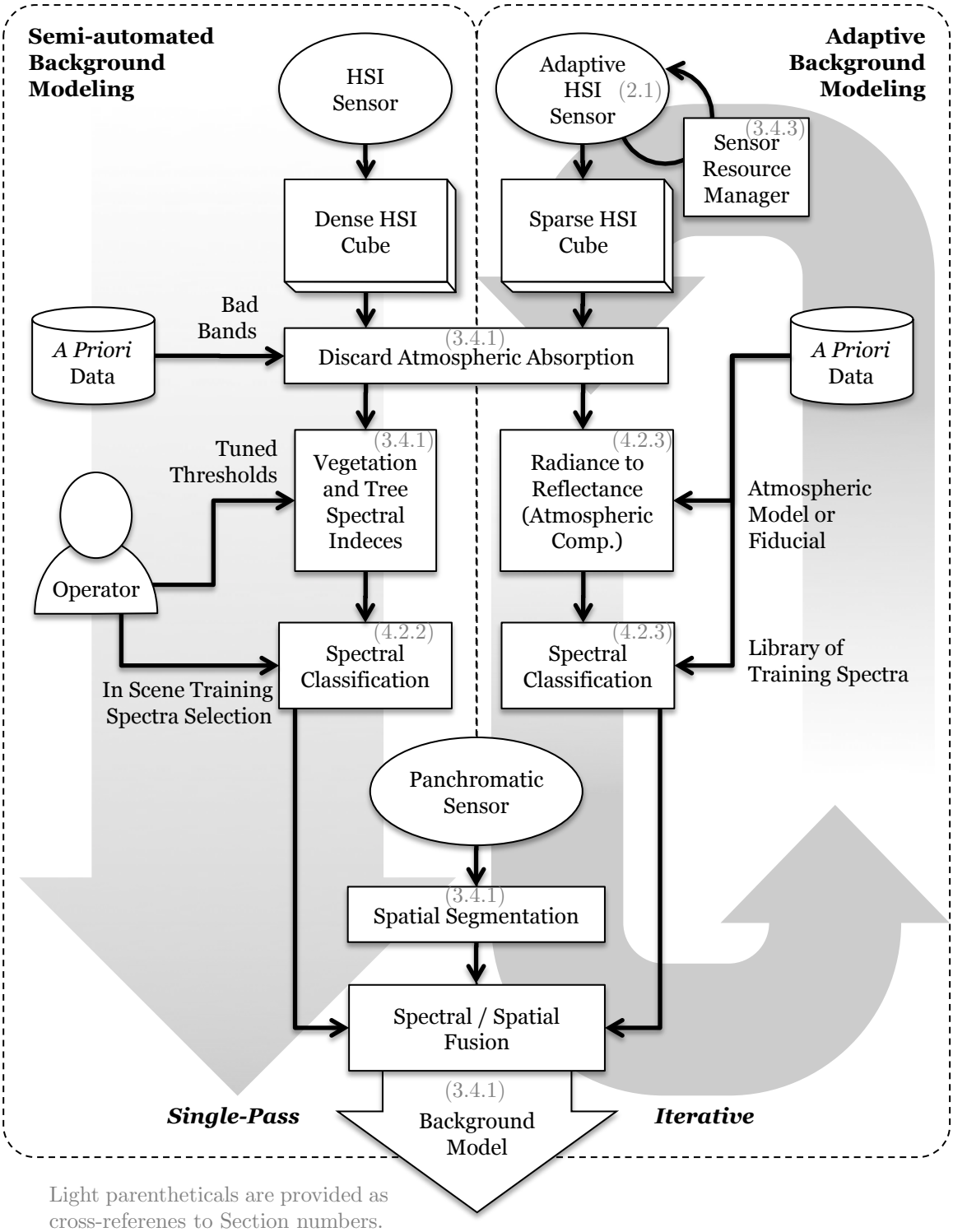


Figure 1.2: An illustration of the two background modeling architectures. The semi-automated architecture is characterized by a single-pass, marginal operator input, and a dense HSI data cube. In contrast, the adaptive architecture is iterative, uses no operator input, and requires an adaptive HSI sensor.

gate tracking performance by 50% in certain operating conditions, and not to harm aggregate performance by any statistical amount in *any* operating condition. This finding, as well as related insight into the behavior of a context-aided system, is a key contribution to the tracking community.

A semi-automated background modeling approach will be shown to qualitatively arrive at a very reasonable background model with minimal operator intervention. A more novel approach – which is adaptive and purely autonomous – will be shown, and is an important contribution of this thesis. A new sensor resource management technique will be presented in support of this adaptive background modeling capability. A case will be made for its zero-latency ability to deliver a model "at-any-time" in contrast to a "just-in-time" technique with much higher latency. This "at-any-time" SRM will be shown to converge to a 66% correct adaptive background model in $\frac{1}{18}^{\text{th}}$ the time of the "just-in-time" approach – a 95% reduction in sensor acquisition time. A hybrid technique will be suggested which transitions from the 66% answer to the 100% answer as soon as it has been fully acquired, resulting in a full-performance steady-state. This SRM is also a novel and important contribution of this thesis.

Finally, the context-aided system will be demonstrated in a high-fidelity tracking testbed. An important finding regarding the impact of context-aiding on single-stage multiple hypothesis tracking systems will be explained and resolved quite successfully. This key finding and solution is an important contribution. The final analysis will show that context-aided tracking improves certain aggregate tracking metrics by 4% relative to a system using uniform background statistics. These metrics will be described as important, but heavily diluted by benign content in the scenario. Another metric will be proposed as the most salient measure of performance gain in the context-aided system. This metric will show a dramatic *30% reduction in error* by the context-aided system relative to the best-performing uniform background statistic system. This is perhaps the most important contribution of the thesis, and demonstrates that context-aided tracking with adaptive hyperspectral data is indeed a viable approach.

1.4 Organization

This document is organized into five chapters. Chapter I serves as an introduction to the work. The problem statement and several key challenges are presented. A brief overview of current system capabilities is given. The contributions of this research effort are listed.

Chapter II reviews background information which is helpful for subsequent developments. Topics include hyperspectral sensing, multiple-hypothesis tracking, context-aided tracking, hyperspectral exploitation, and data synthesis.

Chapter III provides the fundamental theory behind the techniques involved in this study. The multiple hypothesis tracker is analyzed in more detail, with an emphasis on track maintenance and background statistics. Hyperspectral exploitation is presented in the context of deriving scene context from hyperspectral data, to include post-processing and material classification. Next, context aided tracking is developed more fully, with an emphasis on background modeling to derive context dependent tracking statistics. Finally, the notion of adaptive background segmentation with a novel adaptive hyperspectral sensor is introduced, including a brief discussion of the requisite sensor resource manager.

Chapter IV presents the experimental design and results, which are intended to explore the effects of context aided tracking on tracker performance. First, a parametric experiment is presented, in which an extensive Monte Carlo analysis demonstrates the fundamental capabilities of context aided tracking. Next, a full tracking experiment is performed, in which a mature, multiple hypothesis tracking testbed is augmented with context aiding derived in a semi-automated fashion from synthetic hyperspectral data.

A conclusion is given in Chapter V. The effort is summarized, identifying key contributions to the field, and suggesting opportunities for future study. Appendix A includes `Matlab`® code for the parametric experiment, which may be of use for further studies.

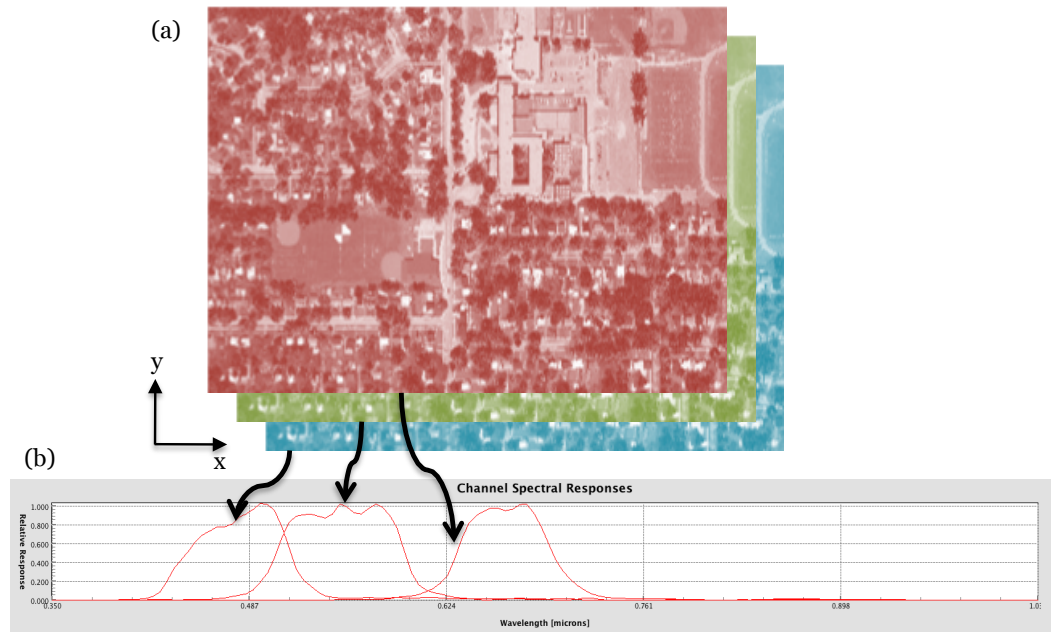
II. Background

2.1 Hyperspectral Sensing

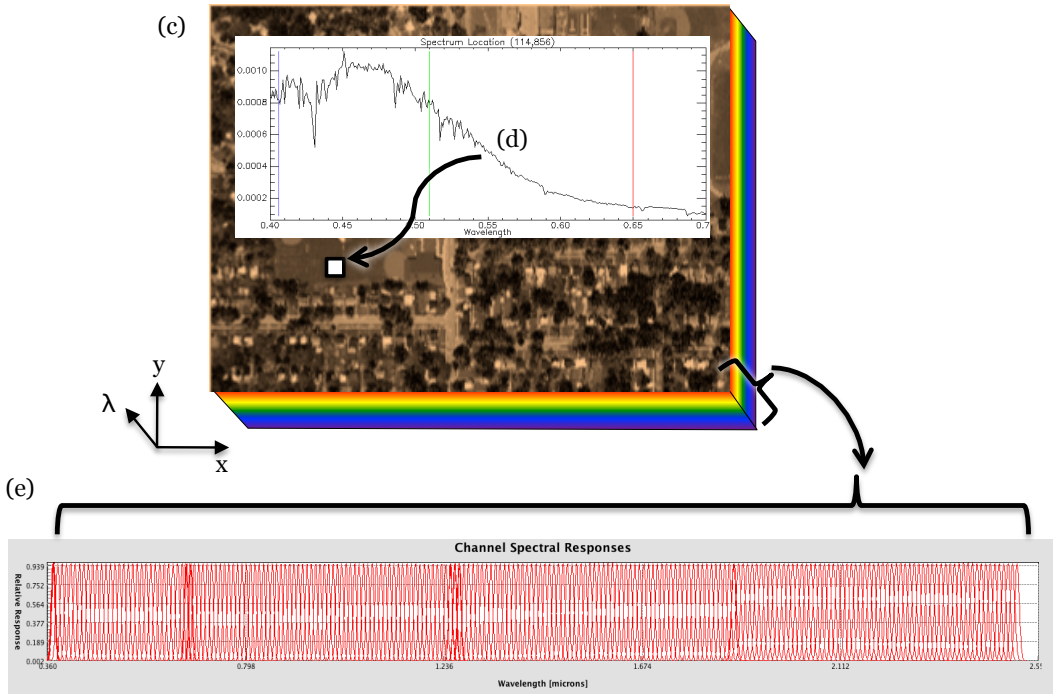
Hyperspectral imaging (HSI) is a sensing technique which measures light in two spatial dimensions, and one spectral dimension divided into N – typically more than 100 – discrete and narrow spectral bands. The resulting spectral image is commonly referred to as a hyperspectral cube. In contrast, the more familiar concept of color imaging generally divides the spectrum of light into three moderately wide bands: red, green, and blue. Panchromatic imaging measures light in a single wide band, forming a familiar grayscale signal. An illustration of HSI is given in Figure 2.1.

The spectral content of HSI data are well suited for material classification, detecting camouflaged targets, and reacquiring previously observed targets. While there are many methods to design hyperspectral imagers, dispersive spectrometers is the type most frequently employed in remote airborne sensors. These instruments take advantage of the wavelength-dependent nature of light undergoing refraction or diffraction, as with a prism or grating. In order to use a two-dimensional electro-optical focal plane array to capture this three-dimensional image, one spatial dimension is generally restricted by a slit-shaped aperture whose orientation is complimentary to that of the dispersion. By mechanically scanning the sensor in the cross-slit direction – as with a moving platform or a tilting mirror – the lost spatial dimension is recovered, albeit at the expense of time. An obvious detriment of this approach is that at any given moment, the field-of-view of the sensor is limited by the narrow slit, and the revisit rate may be quite low. Traditional dispersive HSI is ill-posed for the challenge of remotely tracking moving vehicles in urban environments due in part to this limitation, as well as the high processing burden of full HSI cubes.

It has long been recognized that if the two spatial dimensions of the resulting image are allowed to be sparse, i.e., several discrete objects of interest, then a continuous field-of-view, non-scanned instrument can be realized. For instance, in a multiple object spectrometer (MOS), the slit-shaped aperture is replaced with a mask consisting of an open point for each object to be observed. Constraints on the distribution



3 Broad Spectral Bands (RGB imager)



224 Narrow Spectral Bands (AVIRIS Hyperspectral)

Figure 2.1: An illustration of hyperspectral imaging. In a red/green/blue (RGB) imaging system, there are three component images (a) with spatial dimensions x and y. Each of these images arises from the spectral response of the imager (b) in one of three broad bands: red, green, and blue. In contrast, the output of a hyperspectral imaging system is a cube (c) with three dimensions: spatial x and y, and a spectral dimension λ. For each discrete spatial location in the cube, its hyperspectral signature (d) is a measurement of light along the spectral dimension λ, and is a function of the spectral wavelength measured in many narrow spectral bands (e).

of points and spectral bandpass filters ensure that the spectra from multiple points do not overlap. Historically, MOS instruments have been applied to sensing scenarios with extremely deterministic platform/object relationships, e.g., astronomy.

With the advent of digital micromirror device (DMD) arrays (DMA), the Rochester Institute of Technology Multi-Object Spectrometer (RITMOS) [36] instrument replaces the aperture mask with millions of small mirrors, illustrated in Figure 2.2. This has the advantage of extremely fast mask reorganization, as well as repurposing the light "wasted" by the MOS to capture a panchromatic channel. The layout of the RITMOS imaging elements are illustrated in Figure 2.3. This thesis discusses a notional RITMOS-inspired instrument which is the *Adaptive HSI Sensor* element of the background-modeling architecture in Figures 1.1 and 1.2. As in [45], this instrument will be applied to the urban vehicle tracking problem.

2.2 Multiple-Hypothesis Tracking

Here, a tracking system will be described which forms the *Multiple Hypothesis Tracker* (MHT) [1] element of the system architecture in Figure 1.1. The modern ground-target tracking system employed in an airborne remote-sensing system is responsible for maintaining detailed knowledge of the state of many objects within an area of regard. Practical tracking systems must perform several key functions: filtering, gating, multiple-track to multiple-measurement data association, and automated track initiation/deletion. Tracking literature is replete with treatments on these; [13] is a comprehensive reference and bibliography.

Fundamentally, filtering is the process of deriving optimal estimates of the state of a sensed object given measurements of that object in the presence of process and measurement noise. A vast body of research has been invested into optimal estima-

[1]The tracking community has abstracted the term multiple hypothesis tracker (MHT) to refer to any full tracking system which incorporates the multiple-hypothesis construct for deferred association uncertainty management. Here, the author will attempt to follow this convention. When the association uncertainty management construct is being specifically referenced, the term *MHT construct* will be used.

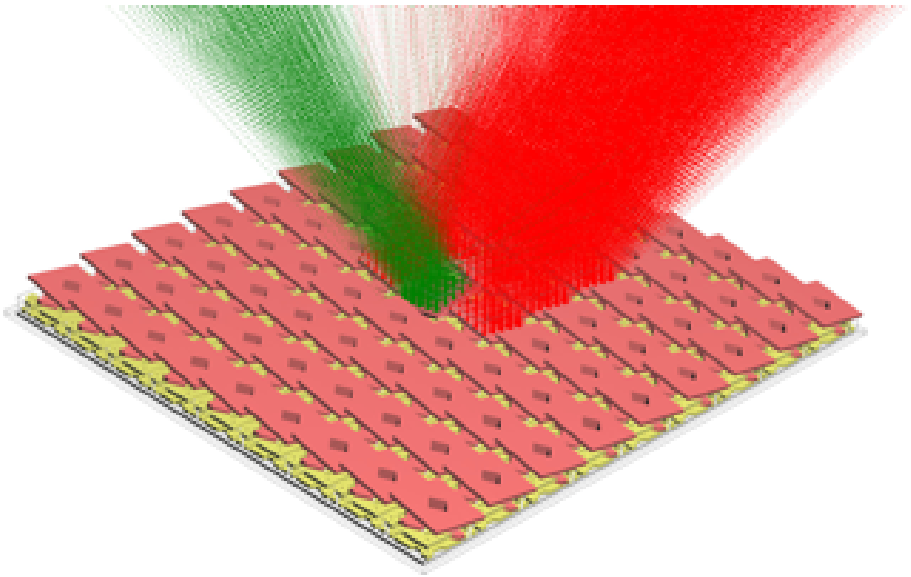


Figure 2.2: An illustration of a small portion of a digital micromirror device array (DMA). Here, all micro-mirrors are set towards a particular path (the righthand side) except for the centermost micro-mirror. Incident light is predominantly steered towards the righthand path, which is illustrated in red for several micro-mirrors near the center. However, light incident upon the centermost micro-mirror is steered towards the lefthand path. Although difficult to discern in this illustration, some very small portion of light rays reflect off of vias or hinge structures between the micro-mirrors and follow the wrong paths. This reduces the imaging performance of the overall instrument.

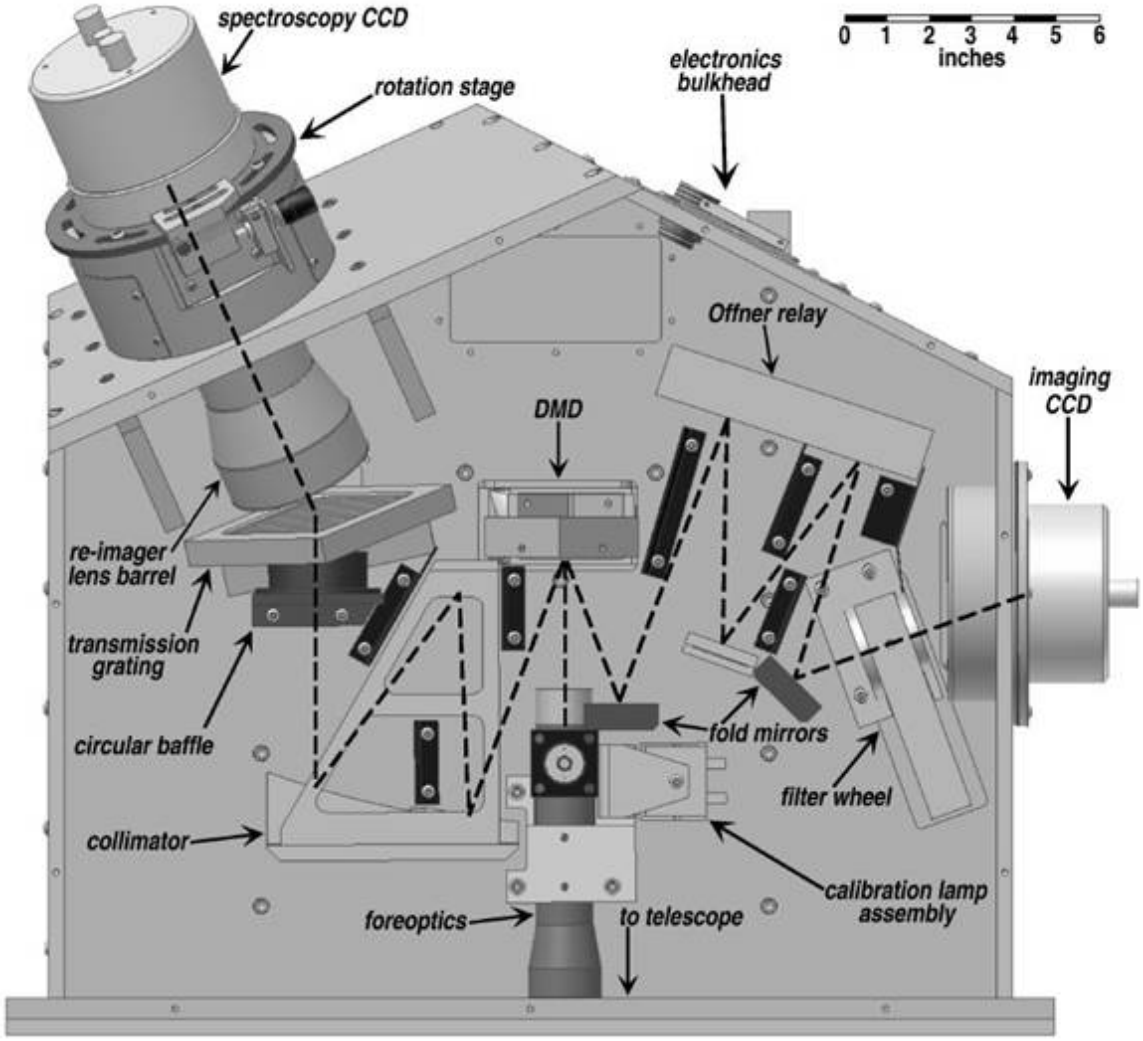


Figure 2.3: The Rochester Institute of Technology Multi-Object Spectrometer (RITMOS), which incorporates two light paths: imaging and spectroscopy. Each pixel is steered towards a light path independently via the digital micromirror device (DMD).

tion and predates modern digital computer systems – indeed, Gauss worked through this problem over 200 years ago [22]. However, the fundamental basis for modern filtering approaches is the Kalman filter [29]. The Kalman filter is an optimal, recursive, minimal-mean-squared-error estimator founded in Bayesian statistics. Many subsequent advances in tracking systems retain Kalman filtering techniques within more complex architectures.

Gating is a data reduction method used to potentially reduce the number of possible track-to-measurement associations in a tracking system. By employing a gating technique, measurements which are highly unlikely to associate with a given track are simply not considered for that association problem. Gating generally begins with determining the distance between an estimated track state and all measurements. Based on straightforward constraints, e.g., maximum target speed – and possibly incorporating the uncertainty of the track state – unlikely measurements are excluded from the gate.

Much of the complexity of the tracker arises from the incorporation of multiple targets. For each measurement-to-track association, there is a resulting cost generally related to some probabilistic distance between the track's predicted state and the measurement. This concept of cost is simply a convenient reversal to that of an association score and arises since cost is analogous to distance. For further convenience, the cost is frequently expressed as a likelihood ratio and then as the logarithm of that ratio. This form permits track costs to be easily composited in case multiple filters are involved (such as a kinematic Kalman filter and a feature filter) and makes recursive cost computation straightforward. Nontrivial composite association events with multiple tracks and measurements – those that consist of missed measurements, false alarms, object entrances/departures, and closely spaced tracks – frequently lack an obvious measurement-to-track assignment solution. Instead, there is competition between tracks and measurements which must be resolved via data association. A set of these associations – a solution – assigns measurements to tracks (taking into account missed detections and new tracks) and has a composite cost accounting for

all tracks, normalized appropriately. Finally, many different solutions are possible and can be ranked according to their composite costs. The global-nearest-neighbor (GNN) solver simply uses the highest ranking of the solutions at each time. Since the second and lower ranked solutions are discarded, this method has little hope of recovering from an association error. The probabilistic data association filter (PDAF) [10] attempts to do somewhat better than the GNN by incorporating several highly likely solutions into a single answer. The joint probabilistic data association filter (JPDAF) [20] is a necessary extension to the PDAF in order to handle the multiple-target case. Using these techniques, a track may be influenced by multiple measurements which fall within its gate. Conversely – and somewhat less intuitively – a measurement may be used simultaneously to influence more than one track.

In contrast, the MHT construct [43] preserves many suboptimal association solutions without combining them. This results in potentially many mutually-exclusive hypotheses whose tracks may disagree with respect to measurement association history. To the extent that the various hypotheses capture different solutions to any given assignment problem, that problem's decision has been "deferred," and its solution is "soft." A particular decision outcome may rise or fall in favor as its associated hypothesis incorporates new information over time. Practicality demands that the weaker hypotheses be pruned away, and only a finite length of history is maintained. Thus, the outcomes of any association problem will eventually reduce to one, which becomes "firm."

Automated track initiation and deletion is a collection of track maintenance algorithms which identify emerging and departing tracks without the need for manual intervention. Initiation is arguably the more difficult of the two problems and hence is frequently divided into two stages: initialization and confirmation. Initialization is the process of bootstrapping raw measurements into newly formed tracks. At its simplest form – and leveraging the capacity of the MHT construct to withstand numerous mutually-exclusive associations – it consists of forming a new track at each measurement. These new tracks are necessarily zero-velocity and highly noisy and

are extended according to normal gating rules at the next frame of data. Notably, the MHT construct may be managing competition between mature extending tracks in contention for the same measurements as the new tracks. Certainly these new tracks will be far too prolific and messy to be reported to an operator, motivating the second stage. The confirmation stage is handled via a variety of techniques. An ad-hoc confirmation approach declares tracks firm once they have attained some number of successful updates, perhaps allowing for a small number of missed detections. If more rigorous *a priori* knowledge of the track statistics are known, a track-cost-based method based on the sequential probability ratio test (SPRT) [57] provides a true analytical solution for track confirmation. The SPRT, if attainable, provides a straightforward threshold against which to test the recursive track cost. It additionally provides a rejection threshold to permanently reject the confirmation of a badly performing track. The following hybrid confirmation approach is loosely based upon the SPRT. The prototype-based confirmation uses the sequential testing structure, but replaces the analytical thresholds with a prototypical track. This prototype track is designed such that it has performed as badly as possible, e.g., missed as many measurements as possible, but is still reasonable to confirm. Once confirmed, a track is then eligible to be presented to the operator or other downstream consumer. It may also receive special precedence in the association logic.

Track deletion is simply the reverse of the confirmation problem, and uses the same family of techniques. A deleted track may be permanently removed from the database and no longer presented to the operator or other downstream processes. Deleted tracks may also be retained in the database for use with track "stitching" techniques, but for the purpose of this research, track deletion is a permanent process.

2.3 Context-Aided Tracking

Research in the field of CAT has incorporated a broad set of techniques to apply background information to the tracking problem. Many elements of a tracking system are eligible for augmentation with scene context. The choice of which element or

elements are to be aided with context is largely driven by the fidelity of the contextual data and the practical limitations of the tracking system.

2.3.1 Detection Masking. In the AFIT thesis [41], target motion detection was performed in persistent, wide-area motion imagery (WAMI) of an urban environment. Due to challenging occlusion and parallax effects from buildings in the post-registration imagery, the majority of the observed motion in such data may not be target related at all. In such an extreme false-detection environment, the entire downstream tracking system is under a heavy computational burden and prone to error. Since GIS road network data is readily available for many urban areas, the author chose a technique in which the known road centerline is dilated to create a curb-to-curb road mask extending over the road network. Given three dimensional building geometries – as is available from light-detection-and-ranging (LIDAR) – coupled with good platform position information, the author was able to create a parallax estimate of the buildings in any observation frame. The union of the negative of the road mask with the parallax mask formed a comprehensive detection mask. This mask served as a straightforward filter applied early in the motion detection stage of the tracker, eliminating the majority of the false detections before they reach the association stage. The remaining detections were on the road and not due to building structure, and hence were more likely to be target related. Clearly a detection-stage application of context is appropriate when the fidelity of the context data is sufficient and the computational benefit is of high importance.

2.3.2 Association and Track Scoring. The multiple-measurement to multiple-track association solution is based upon an ensemble of discrete initialization, update, and missed-detection costs. The formulation of these costs is discussed at length in Section 3.2. These costs are based upon estimates of certain statistics, such as target detectability, and prevalence of false detections. Since these statistics are very likely to be spatially dependent, it stands to reason that the use of scene context may improve their efficacy. This forms a key motivation for this thesis.

2.3.3 Target State Estimation. A popular use of scene context is to estimate hospitability, which is a high-level statistic combining a variety of factors. These frequently include terrain, land-cover, and surface water features. It has classically been used in a relatively low-spatial-frequency manner to cover large areas, e.g., a particular $10km^2$ portion of a military operating theater may be flat with soil suitable for tactical vehicles – hence it is hospitable. In [30], the authors formulate a target probability density function which directly incorporates hospitability into its update function. Road network context is also eligible to be considered in target state estimation. Introduction of such a network causes undesirable nonlinearities if applied directly to the state estimation logic. However, in [42], an elegant solution to this problem is proposed using pseudo-measurements [55]. These are carefully crafted fictitious detections designed to nudge the state estimate towards a road.

2.3.4 Target Dynamics. In [15], CAT is applied to the problem of cooperative user tracking in cellular phone networks. The Euclidean state and dynamics model has been replaced with a road-segment state and along-road dynamics model, resulting in a much lower data transmission burden within the same error budget. This road-constrained motion model technique is also used in "map-matching" applications such as consumer global positioning system (GPS) navigation aides. A noncooperative version of this is used in [14] to data-mine road traffic sensors and infer discrete vehicle tracks through large cities. Inferring plausible target destination from GIS was studied in the AFIT thesis [37], in which prediction time horizons were considered beyond those in which pure target kinematics was useful. The author applied graph-theory – such as Dijkstra's shortest-path algorithm – to the road network in order to predict target destinations. A related concept is that of track stitching, where a track has been terminated and is later tested against a new emerging track for feasible sameness. When the gap in time covered by the track stitch is large, the context provided by the road network may be the most salient measure of feasibility.

2.4 Hyperspectral Exploitation

2.4.1 Scene Classification. The goal of scene classification is to characterize the scene context based on training samples. In training, a model is derived from spectra with known or "supervised" material identity; discrete materials become "classes." In parametric classifiers, this model assumes some underlying density function – usually Gaussian; the quality (and even attainability) of the derived parameters is subject to the population size of the training data and the goodness-of-fit to the assumed distribution. Non-parametric classifiers, such as the generalized-relevance-learning-vector-quantization-improved (GRLVQI) method from [35], relax assumptions of the underlying distribution and tend to tolerate smaller training populations.

The following subsections present several common, yet often effective, parametric classification schemes. These methods can work well in practice, depending on the specific problem.

2.4.1.1 Linear Disciminant Functions. The linear discriminant functions (LDF) assumes each class is distributed normally where each class has a unique mean vector but that all classes have a common covariance matrix. The LDF defines variables for a Gaussian distribution that are mostly "typical". The prior probability and mean of each class are computed via maximum likelihood and are defined as

$$\hat{\pi}_c = \frac{M_c}{M}, \text{ and} \tag{2.1}$$

$$\hat{\mu}_c = \frac{1}{M_c} \sum_{\mathbf{x} \in X_c} \mathbf{x} \tag{2.2}$$

respectively. The variance, on the other hand, is somewhat different. It is defined as the sum of non-normalized outer products of each class normalized by the number of samples less the number of classes. The reason for this modified normalization term is due to the degrees of freedom lost in the computation of each of the C class means. This form of the covariance follows that of the minimum variance unbiased

estimate (MVUE) of the covariance.

$$\widehat{\sum} = \sum_{c=1}^{C} \sum_{\mathbf{x} \in X_c} \frac{(\mathbf{x} - \hat{\mu}_c)(\mathbf{x} - \hat{\mu}_c)^T}{M - C} \tag{2.3}$$

By simplifying and isolating the discriminant for each of the several classes of interest [27], the linear discriminant functions are formed:

$$\delta_c(\mathbf{x}) = \ln(\hat{\pi}_c) - \frac{1}{2}\hat{\mu}_c^T \Sigma^{-1} \hat{\mu}_c + \mathbf{x}^T \hat{\Sigma}^{-1} \hat{\mu}_c. \tag{2.4}$$

A decision is made by choosing the descriminant with the largest response.

2.4.1.2 Quadratic Disciminant Functions. The quadratic discriminant functions (QDF) assumes each class is distributed Gaussian where the mean and covariance of each class is computed from the training samples from the respective classes. This method is effective *if* enough samples exists in each class to ensure an invertible covariance matrix. For hyperspectral imagery, the samples are often too few and the covariance matrix uninvertible.

One derives the QDF in the same manner as LDF. Since the covariance matrices are not the same, they do not cancel in the log likelihood ratio test and the decision is a quadratic function in $\mathbf{x}$. The discriminant is described as:

$$\delta_c(\mathbf{x}) = -\frac{1}{2}\ln|\hat{\Sigma}_c| + \ln(\hat{\pi}_c) - \frac{1}{2}(\mathbf{x} - \hat{\mu}_c)^T \hat{\Sigma}_c^{-1}(\mathbf{x} - \hat{\mu}_c), \tag{2.5}$$

where $|\cdot|$ is the determinant operator. A decision is made by choosing the descriminant with the largest response, just as in LDF [27].

2.4.1.3 Minimum Euclidean Distance Classifier. The minimum euclidean distance (MED) classifier (also called the Minimum Distance classifier in the literature) can be implemented with any number of similiarity measures or metrics, e.g., the Mahalanobis distance. This particular classification method is simple yet

surprisingly effective (see e.g., [35]). It makes the naive assumption that data are distributed $N(\mu_c, I(n \times n))$, where μ_c is the mean of class $c = 1 \ldots C$ and there are C classes.

The MED consists of two steps. The first is to compute μ_c:

$$\hat{\mu}_c = \frac{1}{M_c} \sum_{\mathbf{x} \in \mathbf{X_c}} \mathbf{x}. \tag{2.6}$$

where M_c is the number of samples in class c. The second is to compute the Euclidean distance between each sample in the image and each of the C mean vectors, one for each class:

$$d_c = \|\mathbf{x} - \mu_c\|. \tag{2.7}$$

The mean vector from the class that results in the smallest Euclidean distance is selected as the winner. That is,

$$p = \arg\min_c \; d_c. \tag{2.8}$$

2.4.2 Nonparametric Classifiers. Variants of the neural learning methods of generalized learning vector quantization (GLVQ) by Sato and Yamada [50] have been implemented. The specifics follow that of the generalized relevance learning vector quantization (GRLVQ [26])-improved (GRLVQI) by Mendenhall and Merényi [35]. It offers improvements to GLVQ and GRLVQ by way of the addition of an equal probabilistic learning process presented by DeSieno's in his conscience learning [17] process originally introduced for the unsupervised self organizing map (SOM) due to Kohonen [32]. The methods implemented allows one to switch between the following configurations:

- Baseline GLVQ with conscience learning as incorporated in GRLVQI in [35]

- Baseline GLVQ with conscience learning and the in-class conditional update method described in [35]
- Baseline GRLVQ with conscience learning as incorporated in GRLVQI in [35]
- Baseline GRLVQ with conscience learning and the in-class conditional update method described in [35]

The GLVQ is a supervised learning paradigm that is an advancement of Kohonen's [32] original LVQ's. In particular, the methods employed here are extensions of the so-called LVQ2.1, which is a form of LVQ that works to define decision boundaries by way of a differential shifting strategy. This differential shifting strategy defines an in-class and out-of-class winning prototype vector, and shift both the in-class winner and out-of-class winner in such a way as to *refine* that decision boundary.

2.4.3 Abstracting Geographic Information from Hyperspectral Data. In the AFIT thesis [28], an automated method of abstracting functional GIS information from HSI data is presented. A particular focus of the effort was to provide a capability that did not depend on *a priori* spectral signatures. Instead, the fundamental spectral topology of the scene is learned through a SOM. Key spectral properties of vegetation are applied to segment portions of the SOM such as trees and grass. A human operator is then permitted to further segment the SOM by choosing training spectra from the HSI data. This allows for the identification of functional classes such as roofs, roads, and parking lots. Next, a morphological processing stage operates on the SOM-derived spectral classification. The morphological processing uses heuristics tuned for each class to increase the accuracy of the GIS output.

The ability to abstract GIS information is a fundamental precursor to the use of HSI data for CAT, and will be extended in Section 3.4.1 of this thesis as *background modeling*. In contrast with the previous method, the background modeling technique presented here is concerned with the application of the derived GIS to a specific problem domain: CAT. As such, the focus here is on the estimation of useful background statistics for the tracking system. Some novel capabilities of the previous

effort, e.g., a road-finding method, are not applied here. A significant departure from the previous method is taken here in *adaptive background modeling*, Section 3.4.2. In adaptive background modeling, an emphasis is placed on the use of non-scene specific *a priori* spectral signatures from common spectral libraries. This precludes the need for operator training-spectra selection. Finally, an adaptive HSI instrument is investigated here for applicability to the background modeling challenge. This introduces the possibility of an extremely rapid background modeling capability at some loss of accuracy.

2.4.4 Feature-Aided Tracking. The automated target recognition (ATR) community has pioneered many machine-learning and pattern-recognition methods for identifying targets in largely unconstrained problem spaces. A dominant sensing modality in this research has been HSI, for which mature techniques have existed in the open literature for nearly a decade [44]. The tracking community, meanwhile, has applied ATR techniques to the more constrained problem of FAT. The seminal DARPA programs Video Verification of Identity VIVID [23,24] and NetTrack [3] have demonstrated the viability of FAT in video and radar respectively. The application of FAT techniques to HSI data has traditionally been difficult due to the low revisit rates and processing latencies of HSI sensors. However, emerging sensing technologies [31] and ever-shortening automated exploitation timelines have led to the emerging field of HSI FAT [12, 45].

In the AFIT thesis [54], the author presents a multiple-target tracking system which – much like this thesis – combines HSI and panchromatic sensors to spectrally augment the tracking capability. The architecture of the work included a baseline kinematic tracking system which derived measurements from the panchromatic sensor, and a hyperspectral target classification stage which collected and processed infrequent target spectral-feature measurements. These feature measurements were intended to resolve ambiguities which arise in kinematic tracking systems. In this manner, the track-identification error that occurs during a kinematic track-swap is

no longer viewed as irrevocable, but instead is corrected with HSI target features. The author chose a fuzzy c-means [18] and SOM spectral classification approach, and achieved approximately 30% performance gain from the HSI FAT technique.

While certainly synergistic with the prior work, this thesis explores the application of HSI data to context-aided tracking rather than feature-aided tracking. The HSI FAT study utilized measured HSI signatures in an expedient synthesized tracking study that captured the first-order effects of FAT. A significantly enhanced HSI synthesis capability has been employed here using the Digital Imaging and Remote Sensing Image Generation (DIRSIG) [52], as described in Section 4.2; it incorporates a vastly larger number of targets in a high-fidelity simulation with proper atmospheric and radiative transfer effects. The adaptive background modeling technique in this thesis takes advantage of an emerging class of adaptive HSI sensors. Should such an instrument be applied to the prior FAT study, there is reason to believe that the performance gain may increase. In fact, given the agility of the adaptive HSI instrument, one could certainly imagine that the combination of the HSI FAT and HSI CAT approaches in a single system would demonstrate performance gains greater than the sum of their parts.

2.5 Data Synthesis

As the proposed adaptive HSI instrument has yet to be realized – and is predicated by the implementation of a real-time feedback controller – it is convenient to use synthetic HSI data to develop and evaluate the system. The DIRSIG model is a first-principals-of-physics based tool useful for synthesizing remote HSI data. DIRSIG accounts for object geometry, spectra, and motion; it uses MODTRAN [46] to apply solar radiance and atmospheric transmission effects. Objects within a DIRSIG scene are "painted" according to underlying spectral-reflectance signatures. Frequently, these are the precisely-measured spectra of real-world objects such as grass, asphalt, concrete, and car paint. The DIRSIG implementation used in this research is detailed in Section 4.2.

III. Theory

This chapter will provide the theory behind the effort, including foundational material necessary for this study, as well as novel techniques proposed herein. First, a brief overview of the optimal estimation theory as applied to tracking is given in Section 3.1.

In Section 3.2, the concept of data association will be discussed in detail to motivate the multiple hypothesis tracking system. Here, key statistics of the background environment are applied to the track costing and maintenance problem. When adapted for CAT, this represents a key contribution of this effort.

Following from the treatment of scene classification in Section 2.4.1, a deeper discussion of nonparametric hyperspectral classification will be given in Section 3.3. Here, the theory of the GRLVQI classifier will be studied in detail.

Finally, Section 3.4 will discuss context-aided tracking in detail. This section represents the most fundamental contribution of this thesis to tracking theory. The concept of background modeling will be introduced, which is concerned with forming a functional map of all background elements in the sensed environment. The novel extension of this into adaptive background modeling will be given next. Adaptive background modeling applies the emerging class of adaptive HSI sensors to the CAT problem. The concept of the adaptive sensor resource manager is a necessary extension to such a sensor. The background model is applied to the tracking system through the development of new track costing and maintenance methods, which are a new contribution to the field.

3.1 Tracking Background

A multitude of tracking methods have been developed and extensive literature exists on this subject. The focus here is on two of the typical functions: filtering and track-to-measurement scoring that are prevelant in many of the tracking architectures. Optimal Bayesian filtering is deemed appropriate for the ground target tracking problem, which is widely accepted in the literature [10, 13]. Furthermore, given the use of a nearly constant velocity dynamics model for ground targets coupled with an im-

age based measurement space, a linear filter model can be implemented. The use of a linear filter model simplifies implementation, and is presented here to provide the foundation for the track-to-measurement scoring that plays a major role in the context-aided tracking development. The goal is to estimate the time varying track state x_k using all measurements z_k collected up to the current time index k. The linear Kalman filter assumes the system model can be defined as

$$x_k = F_k x_{k-1} + w(k), \tag{3.1}$$

$$z_k = H_k x_k + v(k), \tag{3.2}$$

where F_k and H_k are the linear dyanamics and measurement matrices, respectively. The dynamics and measurement noises are represented as $w(k)$ and $v(k)$, respectively.

We will denote a Gaussian distribution with mean μ and covariance matrix Σ as $\mathcal{N}(\mu, \Sigma)$. Similarly, evaluation of the same Gaussian density function at x_k will be denoted as $\mathcal{N}(x_k; \mu, \Sigma)$. We assume $w(k) \sim \mathcal{N}(0, Q_k)$, $v(k) \sim \mathcal{N}(0, R_k)$, and $x_0 \sim \mathcal{N}(\mu_0, \Sigma_0)$, where the statistics Q_k, R_k, μ_0, and Σ_0 are known.

The Kalman filter equations are then given by

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1}, \tag{3.3}$$

$$\hat{\Sigma}_{k|k-1} = F_k \hat{\Sigma}_{k-1|k-1} F_k' + Q_k, \tag{3.4}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k(z_k - H_k \hat{x}_{k|k-1}), \tag{3.5}$$

$$\hat{\Sigma}_{k|k} = \hat{\Sigma}_{k|k-1} - K_k H_k \hat{\Sigma}_{k|k-1}, \tag{3.6}$$

where

$$K_k = \hat{\Sigma}_{k|k-1} H_k' S_k^{-1}, \tag{3.7}$$

$$S_k = H_k \hat{\Sigma}_{k|k-1} H_k' + R_k, \tag{3.8}$$

K_k is the Kalman gain, S_k is the covariance of the innovation term, $y_k = z_k - H_k \hat{x}_{k|k-1}$.

Next, the track-to-measurement association score is well defined for Bayesian filtering in terms of likelihood ratios, with the negative-log-likelihood ratio used for anlaytical convenience. The general form of this ratio is given by

$$-\ln\left[\frac{p(D|H_1)}{p(D|H_0)}\right] \tag{3.9}$$

where $p(D|H_i)$ is the *pdf* evaluated with the data D under the assumption that H_i is correct. This form will be utilized next in the discussion on multiple hypothesis tracking.

3.2 Multiple Hypothesis Tracking

The complexity of the data association function grows aggressively with track density, which is pertinent to the urban tracking challenge. As this association is fundamental to the topic of CAT, an association technique much in the spirit of [13] will be presented here. As mentioned in Section 2.2, the MHT construct is a deferred-decision logic applied to the data association function. The MHT is designed to manage uncertainty by making many mutually exclusive "soft" decisions whenever measurement-to-track conflict arises. However, the MHT depends upon the efficacy of the scoring technique. The score of the myriad hypotheses in the MHT at any time follows directly from the costs of the association events within those hypotheses. Therefore, the notion of track costs is both foundational to the MHT system, and a prime opportunity to apply CAT techniques.

Several key background statistics must be introduced and are classically held as constant parameters within the tracker – indeed they can be used to "tune" the system during its development. These are represented in Figure 1.1 as the *Uniform Background Statistics* element of the system architecture. Their counterpart within the CAT system will be described in Section 3.4.4. The probability of detecting an object (making a measurement) conditioned on its presence is P_{D}. There exists some sufficiently small spatial area A which, when observed by the remote sensing system,

may be measured independently from other such regions for the presence of a moving object. Then an expected density of objects per A is β_{NT} for "new track", and β_{FA} for "false alarm".

The track cost, which is inversely proportional to the track score, is defined as a recursively summed negative-log-likelihood ratio $C(k)$. The cost is initialized for new tracks and evolves as the track is updated or misses measurements:

$$C(1) = -\ln\left[\frac{\beta_{\mathrm{NT}}}{\beta_{\mathrm{FA}}}\right] \qquad \text{initialization} \tag{3.10}$$

$$C(k) = C(k-1) + \Delta C(k) \tag{3.11}$$

$$\Delta C(k) = \begin{cases} -\ln\left[1 - P_{\mathrm{D}}\right] & \text{miss} \\ -\ln\left[\frac{P_{\mathrm{D}} p(z_k^j | x_k^i)}{\beta_{\mathrm{FA}}}\right] & \text{update} \end{cases} \tag{3.12}$$

Consider the following illustrative example of a recursive track cost for a track that is initiated, then updated, and then deemed to have a missed detection.

$$C(3) = -\ln\left[\frac{\beta_{\mathrm{NT}}}{\beta_{\mathrm{FA}}}\right] - \ln\left[\frac{P_{\mathrm{D}} p(z_k^j | x_k^i)}{\beta_{\mathrm{FA}}}\right] - \ln\left[1 - P_{\mathrm{D}}\right] \tag{3.13}$$

The pdf $p(z_k^j | x_k^i)$ is representative of the statistical distance between the j^{th} measurement z_k^j and the predicted track location $x_{k|k-1}^i$ of the track T_i at time k. For the case of the Kalman filter defined above, this expression can be analytically described as

$$p(z_k^j | x_k^i) = \frac{e^{(-d^2/2)}}{(2\pi)^{M/2}\sqrt{|\hat{\Sigma}_k|}} \tag{3.14}$$

where M is the dimension of the measurment space and d is the Mahalanobis distance given by $d = y_k' \hat{\Sigma}_k^{-1} y_k$.

The track cost is a useful method of evaluating promising new tracks to confirm, as well as tracks to drop. Track confirmation helps to insulate the user from false tracks. These originate from false detections and tend to be short-lived, cluttering the

presented track picture. This can be particularly troublesome where false tracks occur near valid tracks; these are called redundant tracks and can convey undue ambiguity to the user. Here, a proposed track confirmation threshold $\mathrm{T_{conf}}$ is based upon the cost of a hypothetical track which initializes and then receives $\mathrm{N_{conf}}$ updates. Since the update cost from Equation (3.12) depends upon the probability $p(z_k^j|x_k^i)$ – which is not known for the hypothetical track in question – a related probability $p_{\mathrm{conf}}^{\mathrm{SS}}(z|x)$ is formed from some steady-state benchmark instead. Hence,

$$\mathrm{T_{conf}} = -\ln\left[\frac{\beta_{\mathrm{NT}}}{\beta_{\mathrm{FA}}}\right] - \mathrm{N_{conf}}\ln\left[\frac{P_{\mathrm{D}}p_{\mathrm{conf}}^{\mathrm{SS}}(z|x)}{\beta_{\mathrm{FA}}}\right]. \tag{3.15}$$

All candidate (unconfirmed) tracks are compared to this threshold and are confirmed if and when

$$C(k) \leq \mathrm{T_{conf}}. \tag{3.16}$$

Of course, $C(k)$ contains all history for that track, which is equivalent to allowing tracks unlimited time to become confirmed.

Likewise, a threshold for dropping badly behaving tracks is necessary. Much as in Equation (3.15), a hypothetical track is conceived which has missed $\mathrm{M_{drop}}$ updates out of $\mathrm{N_{drop}}$ observations. This implies $\mathrm{N_{drop}} - \mathrm{M_{drop}}$ updates, which likewise necessitates a probability $p_{\mathrm{drop}}^{\mathrm{SS}}(z|x)$ from a steady-state benchmark. Hence,

$$\begin{aligned}\mathrm{T_{drop}} = &-(\mathrm{M_{drop}})\ln\left[1-P_{\mathrm{D}}\right] \\ &-(\mathrm{N_{drop}}-\mathrm{M_{drop}})\ln\left[\frac{P_{\mathrm{D}}p_{\mathrm{drop}}^{\mathrm{SS}}(z|x)}{\beta_{\mathrm{FA}}}\right].\end{aligned} \tag{3.17}$$

For the new tracks under test in Equation (3.16), $C(k)$ is an appropriate cost; but for mature tracks it may contain a great deal of history and can grow without bound. Since track deletion is intended to represent events which are sudden in nature – e.g., the vehicle has entered a parking garage – a form of track cost with less memory is desireable. Borrowing $\mathrm{N_{drop}}$ from Equation (3.17) to define this window of time, the

test for dropped tracks becomes

$$\bar{C}(k) \geq \mathrm{T}_{\mathrm{drop}} \text{ , where } \bar{C}(k) = \sum_{\tau=k-\mathrm{N}_{\mathrm{drop}}+1}^{k} \Delta C(\tau). \tag{3.18}$$

Notably, while $p^{\mathrm{SS}}_{\mathrm{drop}}(z|x) = p^{\mathrm{SS}}_{\mathrm{conf}}(z|x)$ is a reasonable design decision, each may alternatively be tuned to a desired rate of track confirmation and deletion. Other tuning parameters, $\mathrm{N}_{\mathrm{conf}}$, $\mathrm{M}_{\mathrm{drop}}$, and $\mathrm{N}_{\mathrm{drop}}$ are set according to the user's balanced tolerance for true track confirmation latency, rate of false track confirmation, prevalence of true track premature deletion, and rate of false track deletion.

3.3 Hyperspectral Classification

Based on the prior discussion of parametric versus non-parametric classification methods coupled with the relatively small sample sizes available in the data for the problem of interest, a non-parametric method will be employed.

Classification is the application of supervised machine-learning techniques to optimally assign identity labels to observed objects. While its applications are broad, here classification will be focused on the HSI domain. Furthermore, while the classification of HSI moving-vehicle signatures is of great utility within feature-aided-tracking (FAT) research [12], this paper will focus on scene background classification for CAT.

A general HSI classification architecture begins with data pre-processing and has a training stage followed by a utilization stage. Preprocessing is concerned with transforming the spectral dimension of the data from N bands (in a native radiance space), to an N_{F} dimensional feature space. Although not mandatory, this commonly includes radiance-to-reflectance conversion and/or dimensionality reduction, such that $N_{\mathrm{F}} \ll N$. This reduction arises from the knowledge that, for materials of interest, portions of the N bands are highly correlated. Also, some bands may have very poor signal-to-noise characteristics, e.g., water absorption in the atmosphere, and should simply be dismissed. The GRLVQI is a gradient-descent neural-learning method. As

it trains, it uses differential-shifting to manipulate an abitrary number of prototype vectors within the feature space. This training can become quite computationally expensive, although adaptations in [11] have led to significant improvements in efficiency in the form of the adaptive generalized relevance learning vector quantization improved (AGRLVQI) classifier. Finally, the utilization stage of classification tests unknown signatures against the model and declares the identity of each – or defers in case of low confidence. In parametric classifiers, this test chooses the class that minimizes some statistical distance. In the case of the AGRLVQI non-parametric classifier, this test chooses the class with the closest individual prototype vector.

Here, the AGRLVQI classifier is applied to the urban HSI CAT challenge; the selection is due mainly to its robustness within an autonomous system. All such GLVQI-based paradigms define a cost function D, which is a function of the decision boundary, and is defined as:

$$D = \sum_{m=1}^{M} f(\mu(\mathbf{x}^m)), \tag{3.19}$$

where $f(\cdot)$ is a sigmoid function that takes into account the loss due to each sample, M is the number of training samples and $\mu(\cdot)$ is the missclassification measure defined as

$$\mu(\mathbf{x}^m) = \frac{d^{IC} - d^{OC}}{d^{IC} + d^{OC}} \tag{3.20}$$

where d^{IC} and d^{OC} are the relevance weighted and squared Euclidean distances between the input sample $\mathbf{x}_m$ and in-class winning prototype vector ($\mathbf{w}^{IC}$) and out-of-class winning prototype vector ($\mathbf{w}^{OC}$) respectively.

The loss function in Equation (3.21) is defined as a function of the classification performance as determined by the nearest in-class and nearest out-of-class winner per Equation (3.22):

$$f'(\mu(\mathbf{x}^m)) = f(\mu(\mathbf{x}^m))\left[1 - f(\mu(\mathbf{x}^m))\right], \tag{3.21}$$

$$f(\mu(\mathbf{x}^m)) \quad = \quad \frac{1}{1+e^{-\mu(\mathbf{x}^m)}}.$$

As such, one may consider AGRLVQI a minimum classification error classifier. The relevance-weighted Euclidean distance is defined as

$$d_\lambda \quad = \quad \sum_{i=1}^{n} \lambda_i \left(x_i - w_i\right)^2, \tag{3.22}$$

where λ is a vector $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ that weights each of the n input dimensions i based on their importance for classification. Note that the appropriate superscripts IC and OC are left off for convenience.

The out-of-class winning prototype vector is the prototype vector with a class label *other* than the input sample, that is closest to the input sample using the relevance-weighted Euclidean in Equation (3.22). Updates to prototype vectors are written generally as $\mathbf{w}\left(t+1\right) = \mathbf{w}\left(t\right) + \Delta\mathbf{w}\left(t\right)$. The values of $\Delta\mathbf{w}\left(t\right)$ are defined in [26] as

$$\Delta\mathbf{w}^{IC} \quad = \quad \frac{4\epsilon(t)^{IC} f'|_{\mu(\mathbf{x}^m)} d_\lambda^{OC}}{(d_\lambda^{IC} + d_\lambda^{OC})^2} \Lambda(\mathbf{x}^m - \mathbf{w}^{IC}) \tag{3.23}$$

$$\Delta\mathbf{w}^{OC} \quad = \quad -\frac{4\epsilon(t)^{OC} f'|_{\mu(\mathbf{x}^m)} d_\lambda^{IC}}{(d_\lambda^{IC} + d_\lambda^{OC})^2} \Lambda(\mathbf{x}^m - \mathbf{w}^{OC}), \tag{3.24}$$

where $\epsilon\left(t\right)$ is the learn rate of the training process and Λ is a diagonal matrix with elements consisting of λ_i.

Updates to the relevance vector as presented in [26] is defined as

$$\Delta\lambda_i \quad = \quad -\frac{2\epsilon(t)^{\lambda} f'|_{\mu(\mathbf{x}_i^m)} d_\lambda^{OC} (\mathbf{x}_i^m - \mathbf{w}_i^{IC})^2}{(d_\lambda^{IC} + d_\lambda^{OC})^2} + \frac{2\epsilon(t)^{\lambda} f'|_{\mu(\mathbf{x}_i^m)} d_\lambda^{IC} (\mathbf{x}_i^m - \mathbf{w}_i^{OC})^2}{(d_\lambda^{IC} + d_\lambda^{OC})^2} \tag{3.25}$$

It is important to note that the updates provide in Equations (3.24) and (3.23) and in Equation (3.25) occur with the *current* values of the weights and prototype vectors.

That is, the values at time t are used, not the values at time $(t+1)$. For classification, one simply assigns a label to the input sample $\mathbf{x}_m$ that is the class label of the closest prototype vector in a Euclidean sense.

3.4 Context Aided Tracking

The premise of CAT is that applied knowledge of a vehicle's environment may improve the performance of the tracker. This knowledge may be available as prepared data in a geographic information system (GIS) or inferred real-time from the surveillance data itself. The later is attractive when GIS data are outdated, denied, or difficult to co-register with the surveillance imagery.

This section will first describe an offline method of inferring context which assumes availability of full-scene HSI data and modest operator intervention. Then the novel, online, adaptive method will be introduced and applied to adaptive HSI sensing. Finally, the background statistics will be directly incorporated into the MHT to provide an innovative method for context-aiding.

3.4.1 Background Modeling. The precursor to CAT is the background model, a spatial map of the scene materials which are functionally relevant to the tracking algorithm. In the system architecture of Figure 1.1, this process corresponds to the *Semi Automated Background Modeling* block. Here, a technique similar to [28] is used to convert HSI data into such a model. A specific synthetic scene generated by DIRSIG is used for the discussion that follows. As such, specific numbers are stated such as the number of hyperspectral bands, number of class models, etc. to illustrate the concepts. A HSI cube of the scene is obtained and rectified such that the mission imagery can later be registered to the resulting model. Radiance (L) to reflectance (ρ) conversion is not essential here, as no reference spectra will be incorporated into this method and atmospheric effects are assumed constant across the scene. However, several bands with moderate to severe atmospheric H_2O absorption are discarded, e.g., 0.93μ. Figure 3.1 illustrates common atmospheric absorption bands in the solar

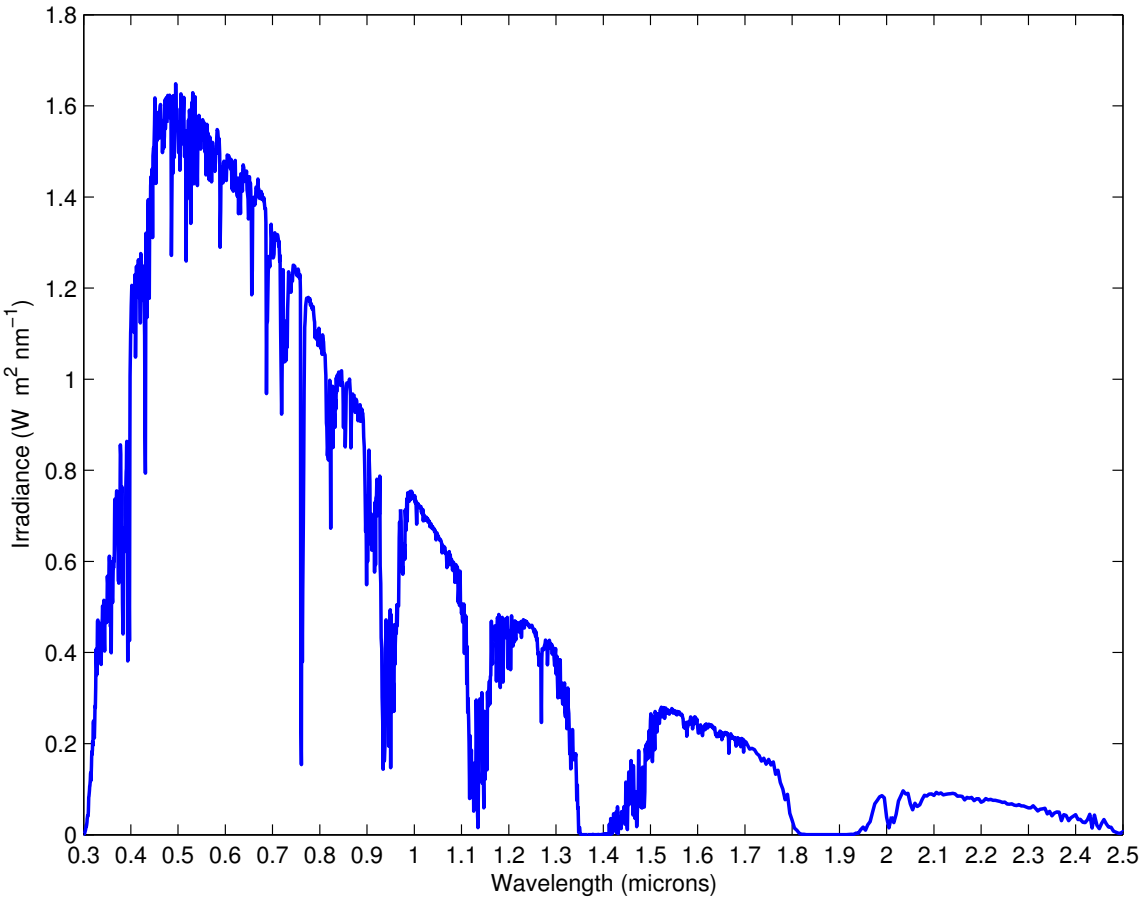


Figure 3.1: The measured solar irradiance at the earth's surface. Notably, this differs from the expected blackbody radiation curve due to atmospheric absorption in certain portions of the spectrum. Source: ASTM G173-03 [2].

irradiance measured at the earth's surface. First, the well known normalized difference vegetation index (NDVI) [56] is used to detect pixels dominated by vegetation:

$$\frac{L_{0.86\mu} - L_{0.66\mu}}{L_{0.86\mu} + L_{0.66\mu}} \geq 0.18 \ , \tag{3.26}$$

where the wavelengths and threshold are manually tuned for a given scene, but are in the range of commonly accepted values.

Next, an empirically derived tree index is used to determine which pixels *among those passing the NDVI test* are dominated by tree leaves:

$$\frac{L_{0.86\mu} - L_{0.78\mu}}{L_{0.86\mu} + L_{0.78\mu}} \geq 0 \ , \tag{3.27}$$

and assuming the remaining pixels to be grass. This index is also tuned for a given scene, and may not hold for scenes with different tree and grass species. With explicit dimensionality reduction (for example, from $N = 60$ to $N_{\mathrm{F}} = 2$), these index methods exploit convenient material properties in an effective and computationally inexpensive way.

Subsequent spectral processing focuses on the much more challenging task of classifying remaining scene elements: roads, water, and building materials. An operator manually identifies subclasses of materials and marks small training regions within the cube for those materials. A functional class, e.g., road, may have many subclasses such as concrete, asphalt, and weathered asphalt; this is necessary to keep the spectral-feature-space variances low. In this experiment, there are 15 subclasses. Ten of these subclasses account for various asphalt-shingle and gravel roof treatments; four subclasses account for road surfaces. The AGRLVQI classifier algorithm is trained on this population of spectral samples with corresponding subclass labels. Next, the resulting classifier model is used to assign labels to all non-vegetation pixels in the cube. The subclass labels are then discarded and replaced with their functional parent class labels. Finally, the fusion of the results from the vegetation indeces and AGRLVQI classifier becomes the initial spectral background model for the scene.

Thus far, spectral domain information has been exploited for each pixel independently. This is apparent in the presence of anomalous "speckles" in the background model. These small features are frequently the result of classification errors, perhaps due to scene materials which were excluded from the training data. At best, these are of no use for the functional model. Clearly, there is further salient information in the two spatial dimensions of the cube, e.g., texture and edge contrast. Recognizing

this, a spatial segmentation technique is applied to each band. Similar to a raster-scan flood-fill operation, this replaces each pixel's radiance with that of its same-band spatial-neighbor if their radiances are within some small threshold. After this replacement, many neighboring pixels will have identical values for some majority of bands in this "stack of bands" – these are called *segments*. Every pixel now belongs to a single segment (or is a singleton segment itself). Those segments with fewer than some number of members is then discarded by assigning it to a larger neighbor segment. Finally, this spatial segmentation is fused with the initial spectral background model: for each segment, all member pixels vote for a material label based on the corresponding cell in the spectral background model. This then forms the final background model.

3.4.2 Adaptive Background Modeling. The background modeling approach discussed above is effective, but comes with a high cost: the requirement for pre-tracking-time HSI acquisition, and an offline human-in-the-loop processing stage. A new approach is considered here, and represents the *Adaptive Background Modeling* block in the system architecture of Figure 1.1.

The emergence of adaptive HSI sensors – such as a RITMOS-inspired DMA-based instrument – has provided a potential path for improvement over the prior method. This instrument collects a full-frame panchromatic image at each step in time. A minority of pixels can be excluded from the panchromatic image on command, via flipping micro-mirrors in the DMA, and reflected into a spectrometer. One concept of employment for such a sensor is to begin the surveillance mission by investing the time to scan a dense HSI cube solely for the sake of background modeling. While this is practical and provides a *just-in-time model*, it is arguably not optimal: more HSI pixels are being collected than necessary. Also, in some cases a rapidly moving field-of-view is desirable or unavoidable, leaving no time for a dense cube. What is needed is an adaptive *at-any-time model* which initializes with as few HSI pixels as possible, yields the best model possible at any time, and converges on the model

afforded by a dense acquisition. Here, a notional DMA-based MOS HSI sensor is applied to the remote ground-vehicle CAT problem and forms the basis for *adaptive background modeling*.

The typical background modeling process in Section 3.4.1 is modified into a form of Bayesian inference [13] as follows. Introduce an *a priori* library of labeled spectral signatures representing N_ϕ functional background material classes ϕ. As in Section 3.4.1, allow the underlying library representation to contain subclasses as necessary. Use the library to perform an offline training process to prepare the NDVI and tree index wavelengths/thresholds, and to train the AGRLVQI classifier model, such that HSI pixels can be labeled.

The online portion begins with a full-frame panchromatic radiance image divided into regions of similar intensity via the spatial segmentation process previously described. Clearly, since this operates on a single (panchromatic) band, the ability to distinguish regions with homogenous materials will be diminished versus the "stack of bands" approach. However, this segmentation provides an initial guess at the background model, with the caveat that each segment lacks a material label.

The remainder of this discussion applies to each segment in parallel, but for brevity, no segment index has been added to the notation. Assume a sequence $\mathbf{Z}$ of labeled, single-pixel hyperspectral observations $\mathbf{z}$ intersecting some segment. Note that the library and observations are unlikely to lie in a consistent radiance space. Radiance-to-reflectance transformation is suggested but beyond the scope of this paper; see [51] for a survey of techniques. One, many, or none of these observations may arrive at each time k. In order to strengthen a claim of independence, we require the observations to be sufficiently separated spatially *or* temporally within the segment. Define the probability that the segment has the functional material label ϕ_i upon incorporation of the 1^{st} through n-th observations as $p(\phi_i|\mathbf{Z}_n)$. At some loss of

optimality, an assumption of uniformly distributed priors is made:

$$\forall i : p(\phi_i|\mathbf{Z}_0) = \frac{1}{\mathrm{N}_\phi}. \tag{3.28}$$

Now define the transitional probability of receiving a specific sequence of observations $\mathbf{Z}_n$ conditioned upon a true class identity ϕ_i as $p(\mathbf{Z}_n|\phi_i)$, and due to independence of the observations

$$p(\mathbf{Z}_n|\phi_i) = \prod_{j=1}^{n} p(\mathbf{z}_j|\phi_i). \tag{3.29}$$

Note that an estimate of $p(\mathbf{z}|\phi)$ is empirically available as a consequence of the *a priori* library via a confusion matrix analysis. Now process the segment's observations in order, recursively updating $p(\phi_i|\mathbf{Z}_n)$ according to

$$p(\phi_i|\mathbf{Z}_n) = \frac{p(\mathbf{z}_n|\phi_i)p(\phi_i|\mathbf{Z}_{n-1})}{\sum_j p(\mathbf{z}_n|\phi_j)p(\phi_j|\mathbf{Z}_{n-1})}. \tag{3.30}$$

For the downstream CAT functionality, it is necessary to assign a single functional material label to each segment. This time-varying label is simply the maximum *a posteriori* label

$$\phi_{\mathrm{MAP}}(k) = \underset{i\in[1,\mathrm{N}_\phi]}{\arg\max}\, p(\phi_i|\mathbf{Z}_{n(k)}) \ , \tag{3.31}$$

and when determined for each segment, becomes the *adaptive background model*.

3.4.3 Sensor Resource Manager. In the background-modeling architecture of Figure 1.2, the *Sensor Resource Manager* is the system component which commands the adaptive HSI sensor in a feedback fashion. As mentioned in Section 2.1, MOS spectrometer instruments restrict the quantity and placement of the HSI pixels which can be acquired at any moment. As a simple example, if two micro-mirrors from the same column and nearby rows were steered towards the spectrometer at the same time, their dispersed radiance would likely overlap, destroying both signatures. Hence, a *sensor resource manager* (SRM) is now conceived. Let $u(k+1|k)$ be the utility of adding a new observation $\mathbf{z}_{n(k)+1}$ at a future time $k+1$ for some segment.

The SRM must then allocate spectral observations, within the constraint, in order to maximize the summed utility. For the simplified constraint of at most one spectral measurement per column on the DMD array, this allocation is simply that which selects the maximum value of $u(k)$ per column when all region's $u(k)$ values are mapped in a row/column space defined by the DMD array.

Three versions of $u(k)$ are given here, and their performance will be compared in Section 4.2.3. First, define the time-varying *entropy* of a segment – a measure of disagreement between the segment's observations – as

$$h(k) = -\sum_{i=1}^{\mathrm{N}_\phi} p(\phi_i|\mathbf{Z}_{n(k)}) \ln \left[p(\phi_i|\mathbf{Z}_{n(k)})\right] . \tag{3.32}$$

The first and most intuitive form of $u(k)$ is simply the entropy itself

$$u_{\text{entropy}}(k+1|k) \triangleq h(k). \tag{3.33}$$

In this formulation, the SRM will clearly devote the most HSI acquisitions to regions with the highest entropy. Since the observations $\mathbf{Z}_n$ for a segment are subject to measurement noise and classification error, there is hope that as $n(k)$ increases, $h(k)$ will decrease. There is, however, cause for concern regarding regions which are difficult to classify, i.e., high $h(k)$ despite many measurements (high $n(k)$). The actual utility of allocating additional measurements to such regions may be quite low – a version of diminishing returns.

The second form of $u(k)$ is designed to mitigate this effect by normalizing the entropy according to the number of measurements

$$u_{\substack{\text{norm} \\ \text{entropy}}}(k+1|k) \triangleq \frac{h(k)}{\ln\left[n(k)+1\right]}, \tag{3.34}$$

where the +1 term accounts for the diminishing utility of the potential new measurement at time $k+1$, and taking the natural log of the denominator puts it on the same logarithmic scale as the entropy in Equation (3.32).

A third form for $u(k)$ is a type of null SRM, in which $p(\phi_i|\mathbf{Z}_n)$, $\phi_{\text{MAP}}(k)$, and $h(k)$ are computed as normal but *not* fed-back into $u(k)$. Lacking that information, the SRM defaults to uniform random behavior

$$u_{\text{random}}(k+1|k) \triangleq U(0,1), \tag{3.35}$$

where the range $[0,1]$ is arbitrary but consistent. This random SRM is motivated by research in the field of compressive sampling – which is closely related to the adaptive sensing method here – in which random measurements are frequently the best method one can employ [47]. Of course, such an SRM that relies on digital pseudorandom number generation techniques has a very tangible computational benefit over those employing entropy analysis.

Regardless of the chosen form for $u(k)$, the utility for CAT could certainly be combined with other utility functions within the system. In a FAT/CAT system there would be a competing desire to measure the signatures of moving vehicles. A system with stationary-target ATR might also have a utility function for scanning the scene to discover new targets. The fusion of multiple utility functions in a similar system was treated in [53].

3.4.4 Adaptive Background Statistics. Traditional uniform background statistics of the environment (P_{D}, β_{NT}, and β_{FA}) were presented in Section 3.2 and are so abstract as to be difficult to estimate. As such, they often degenerate into physically meaningless – albeit important – tuning variables. Part of this difficulty arises from the application of these statistics as uniform values. The CAT paradigm, however, holds the background statistics as spatially dependent. Recognizing that $P_{\text{D}}^{\text{CAT}}$, $\beta_{\text{NT}}^{\text{CAT}}$, and $\beta_{\text{FA}}^{\text{CAT}}$ are difficult to know directly, context information is used to

Table 3.1: Background Statistics

Material	P_{D}	β_{NT}	β_{FA}
Road	0.99	1×10^{-2}	1×10^{-2}
Grass	0.99	1×10^{-4}	1×10^{-2}
Shadowed road	0.6	5×10^{-2}	1×10^{-2}
Tree canopy	1×10^{-2}	1×10^{-5}	1×10^{-2}
Rooftop	1×10^{-4}	1×10^{-8}	1×10^{-2}
Water	1×10^{-4}	1×10^{-8}	1×10^{-2}
Uniform Background Statistic	0.97	1×10^{-4}	1×10^{-2}

heuristically estimate them. Here, context is primarily a question of the functional material composition of the scene – a background model – and is developed as in Sections 3.4.1 and 3.4.2. The first-order effect on $P_{\mathrm{D}}^{\mathrm{CAT}}$ is degree of occlusion. Assuming an airborne sensor, materials which tend to obscure ground vehicles will result in lower $P_{\mathrm{D}}^{\mathrm{CAT}}$. Tree canopies have varying density, but in urban environments, dense and multi-layer canopies are rare. In oblique viewing geometries of urban scenes, vehicles may appear to be behind buildings and rooftops. Further assuming a passive imaging sensor, solar illumination also has a strong effect on detectability, which varies spatially due to shadowing. The statistic $\beta_{\mathrm{NT}}^{\mathrm{CAT}}$ is treated here as a hybrid measure of detectability and hospitability (where ground vehicles can travel). While the statistic $\beta_{\mathrm{FA}}^{\mathrm{CAT}}$ may indeed vary according to material, this would be due to subtleties within the motion detection algorithm. Here, $\beta_{\mathrm{FA}}^{\mathrm{CAT}}$ is held constant. An empirical analysis of real remote sensing data processed with a typical motion detection algorithm has led to the values in Table 3.1.

These statistics are formed into a spatial map, and must be drawn according to location. For $\beta_{\mathrm{NT}}^{\mathrm{CAT}}$ and $\beta_{\mathrm{FA}}^{\mathrm{CAT}}$, the location is intuitively based upon where the measurement z_k^j falls. For $P_{\mathrm{D}}^{\mathrm{CAT}}$, there is some question as to whether this is based upon the location of the measurement z_k^j, the predicted track state $\hat{x}_{k|k-1}^i$, or the posterior track state $\hat{x}_{k|k}^i$. The predicted track state is always available and is a good choice, whereas the measurement is meaningless in the "miss" case of Equation (3.12).

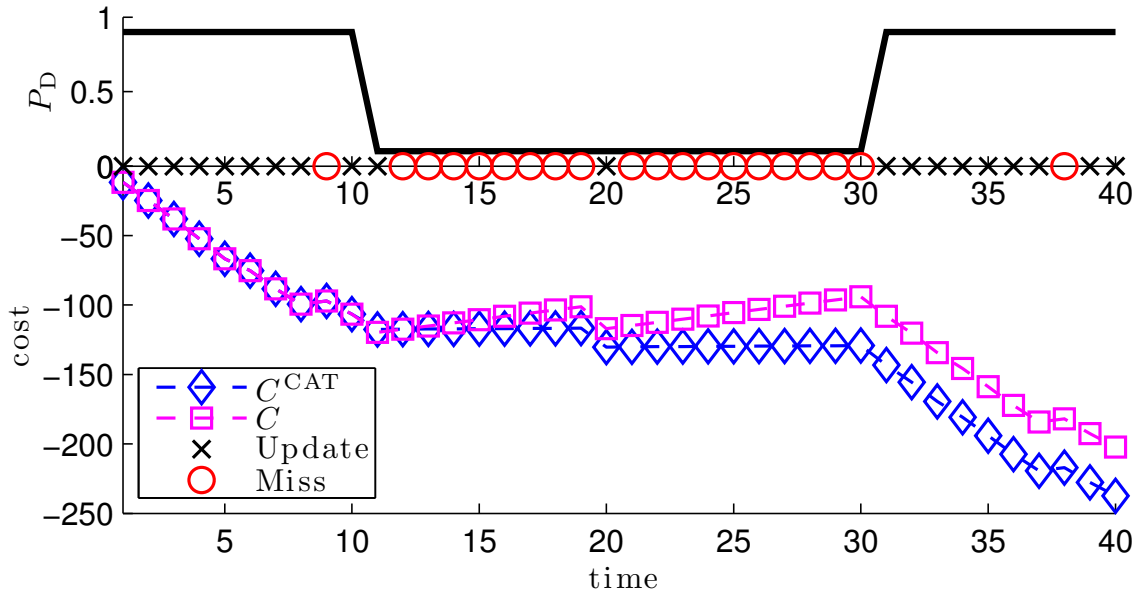


Figure 3.2: An illustration of track cost. This simulated track travels through a high P_D region, a low P_D region, and finally emerges into a high P_D region. The cost C based on uniform background statistics heavily penalizes the track for missed measurements within the low P_D region. The cost C^{CAT} based on context-aided statistics assigns a more reasonable cost to the track in the same low P_D region.

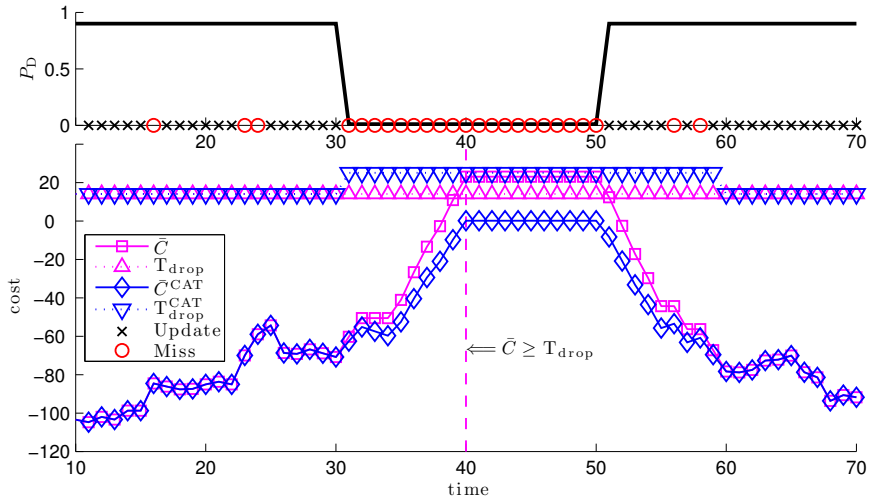
The posterior track state is equivalent to the predicted state in the "miss" case, but subtly different in the "update" case. The posterior represents the optimal estimate of the track at that time, and in this work serves as the reference location for P_D^{CAT}. These statistics then lead to the context-aided track cost C^{CAT} and windowed cost $\bar{C}^{CAT}$. The significance of the difference between C (cost based on uniform background statistics) and C^{CAT} (cost based on context-aided statistics) is readily apparent for tracks traveling through a diverse background. Illustrated in Figure 3.2, C^{CAT} is able to provide a more accurate assessment of track health under certain circumstances.

Also of concern are the context-aided thresholds T_{conf}^{CAT} and T_{drop}^{CAT}, which are based on hypothetical tracks. In the case of confirmation, the hypothetical β_{NT}^{CAT} and β_{FA}^{CAT} are located by the initializing measurement of the track under test. However, N_{conf} updates are assumed to have occurred – since they are hypothetical, their positions are indeterminate, making the selection of P_D^{CAT} questionable. Reasonable

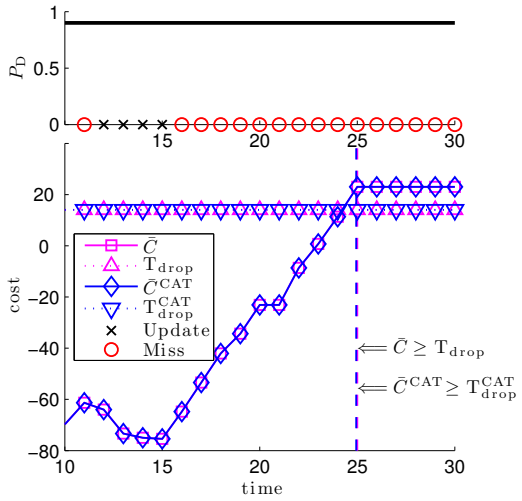
choices include $P_{\mathrm{D}}^{\mathrm{CAT}}$ from the location of the most recent posterior of the track under test, the average $P_{\mathrm{D}}^{\mathrm{CAT}}$ of that track relative to its window $\mathrm{N}_{\mathrm{conf}}$, or uniformly random draws of $P_{\mathrm{D}}^{\mathrm{CAT}}$ throughout that window. An empirical analysis suggests that another more aggressive confirmation strategy reduces the confirmation time of real tracks without a marked increase in false track confirmation: namely the minimum $P_{\mathrm{D}}^{\mathrm{CAT}}$ of the track under test within its window $\mathrm{N}_{\mathrm{conf}}$. Likewise, $\mathrm{T}_{\mathrm{drop}}^{\mathrm{CAT}}$ is based on a hypothetical track with indeterminate locations. Again, the same potential strategies exist for selection of $P_{\mathrm{D}}^{\mathrm{CAT}}$ in the deletion test. Empirical analysis suggests that the minimum within the window yields a balanced but guardedly conservative track drop threshold. Two notional tracking cases have been simulated to validate these threshold choices and to further motivate the benefit of CAT versus uniform background statistics. These cases are illustrated in Figure 3.3.

Case I. A true track, which transitions from a high $P_{\mathrm{D}}^{\mathrm{CAT}}$ region into a low $P_{\mathrm{D}}^{\mathrm{CAT}}$ region and back again. This succinctly describes a primary benefit of CAT: a reluctance to drop tracks known to exist where they are less detectable. An additional benefit is to smoothly transition the drop behavior as the detectability begins to improve. An example of poor transition behavior, which has been resolved by the minimum-windowed method, is the disproportionate drop tendency given a single missed measurement as $P_{\mathrm{D}}^{\mathrm{CAT}}$ rises. An illustration of this case is given in Figure 3(a). The CAT method successfully maintains track throughout this scenario, while the uniform method prematurely drops the track ten frames into the outage.

Case II. A false track, formed entirely of false alarms occurring within a high $P_{\mathrm{D}}^{\mathrm{CAT}}$ region. Although erroneous, this type of event is possible under certain circumstances. Ideally, this track will drop as rapidly as possible. This represents a sort of "control experiment," and although CAT is not equipped to hasten the drop in this case, it preferably should not prolong it. An illustration of this case is given in Figure 3(b). Both the CAT and uniform methods drop the track ten frames after the final false alarm measurement.



(a) Case I



(b) Case II

Figure 3.3: An illustration of track drop-thresholds in simulated data. In (a), the track is following on object which leaves a high P_D^{CAT} region, travels through an occluding low P_D^{CAT} region, and finally re-emerges into a high P_D^{CAT} region. Notably, the uniform cost $\bar{C}_{drop}$ exceeds the uniform drop threshold T_{drop} at frame 40, resulting in track loss. However, the CAT cost $\bar{C}^{CAT}_{drop}$ decreases, and the CAT drop threshold T^{CAT}_{drop} increases during the occlusion. This behavior makes track loss much less likely, and the track is maintained throughout the scenario. In (b), a false track has formed on several correlated false alarm measurements in a high P_D^{CAT} region. The CAT and uniform methods have the same behavior in this scenario, dropping the track ten frames after the final false measurement. This suggests that CAT should not penalize performance in similar cases.

IV. Experimental Design and Results

Here, two experiments are described in which the effects of the proposed CAT system are evaluated. First a parametric experiment is given. While it makes many simplifying assumptions about the underlying environment, it supports rapid testing of a broad parameter and operating-condition space. Second, a high-fidelity, remote-sensing vignette is synthesized and exploited. It processes rendered imagery with a full multiple-target, multiple-hypothesis context-aided-tracking testbed.

4.1 Parametric Experiment

In order to isolate the effects of background statistics on track maintenance performance, a simple parametric experiment has been designed. A sequence of notional observation frames is conceived and processed by a score-based track-maintenance algorithm as in Section 3.2. Here, target arrivals and departures, background elements causing measurement occlusion, and measurement performance are drawn randomly from appropriate underlying distributions. Besides scoring, all other tracker elements – e.g., data association, filtering, and the effects of hypothesis formation – are also replaced with parametric simulators. Notably, the concept of time has been abstracted into frame counts, and frame-rate is neither given nor needed. Also, dimensionless units are used for certain densities. Scaling coefficients have been empirically determined and applied as needed. These simplifying assumptions cause no loss of generality in the results. A suite of `Matlab`® functions has been created for this purpose, and is given in Appendix A. Each run of the simulator produces a sequence of observation frames, and contains exactly one target (except for the null case in which a target never arrives). The target arrival and departure within the sequence is randomly defined, as is the occurrence and duration of occlusions. The extrinsic parameters which define the simulation are given in Table 4.1. These parameters are related to the *truth* of the simulation. The intrinsic parameters which define the exploitation of the simulation are given in Table 4.2. These are *tuning* parameters. For convenience, the parameter names match those in the `Matlab`® source code of

Table 4.1: Extrinsic simulation parameters for the parametric experiment. Parameter names match those in Appendix A. Example values represent baselines within the experiment.

Element	**Distribution**	**Parameter(s) = e.g.**
Number of frames	scalar constant	time = 200
Occlusion arrival	geometric, based on per-frame probability of arrival	pOcclusion = 0.01
Occlusion duration	uniform	minOcclusionDur = 1 maxOcclusionDur = 20
Target arrival	geometric, based on per-frame probability of arrival	pTgtArrival = 0.1
Target departure	geometric, based on per-frame probability of departure	pTgtDeparture = 0.005
Kinematic steady state negative log cost	normal	kssMean = -5 kssVar = 4
Unoccluded measurement	Bernoulli	nominalClearPdTrue = 0.95
Occluded measurement	Bernoulli	nominalOccludedPdTrue = 0.05
False measurement	Bernoulli	pFalseAlarm = 0.05

Appendix A. With this parametric framework, a broad set of operating conditions can be feasibly tested in a Monte Carlo fashion.

A single instantiation of the parametric test requires approximately one second of computation time on a modern personal computer. An illustration of a single instantiation of the parametric experiment is given in Figure 4.1, and will be described here to provide additional insight into the experiment. In this test, a target arrives shortly after the start of the test and remains for approximately 100 frames of time. The simulation generated two occlusions, during which nearly all measurements were lost. This particular instance of the test was setup for uniform background statistics

Table 4.2: Intrinsic tuning parameters for parametric experiment. Parameter names match those in Appendix A. Example values represent baselines within the experiment.

Element	**Parameter(s) = e.g.**
Number of updates in prototypical confirmation track	Nconf = 5
Window length of prototypical dropping track	Ndrop = 10
Number of misses in prototypical dropping track	Mdrop = 5
Coefficient of confirmation (lower confirms sooner)†	confirmFactor = 5
Coefficient of drop (lower drops sooner)†	dropFactor = 5
Assumed ratio of false-alarm density over probability of detection	BetaFA_multiplier = 1×10^{-2}
Mode selection (uniform vs. non-CAT)	CAT = true | false
Assumed uniform probability of detection	unif_Pd = 0.97
Assumed uniform new-track density	unif_BetaNT = 1×10^{-2}
Assumed CAT probability of detection when unoccluded	CAT_Pd_clear = 0.95
Assumed CAT probability of detection when occluded	CAT_Pd_occluded = 0.05
Assumed CAT new-track density when unoccluded	CAT_BetaNT_clear = 1×10^{-2}
Assumed CAT new-track density when occluded	CAT_BetaNT_occluded = 1×10^{-5}

† These coefficients are intermediate tuning parameters used solely to derive $p^{\mathrm{SS}}_{\mathrm{conf|drop}}(z|x)$.

rather than CAT. A track was initialized and confirmed, and successfully coasted through the first occlusion, which was relatively short. During the second occlusion, however, the track was dropped. A new track was initialized and confirmed shortly after the second occlusion completed and measurements resumed. After the true target departed, the system was somewhat delinquent in deleting the track. This is due to a series of false alarms which caused the track to persist before finally being deleted.

The goal of the parametric experiment is to collect a statistically significant population of results while varying certain aspects of the extrinsic or intrinsic parameters. Any across-the-board or conditional performance changes due to CAT should emerge. Since the combinatorics of the parameters make comprehensive testing impractical, a series of tests is defined in which one or two parameters are adjusted at a time.

4.1.1 Metrics for the Parametric Experiment. The following is a set of well-known tracking multi-target metrics which have been identified as most likely to demonstrate the effects of CAT. As this parametric experiment lacks some complexity of a full tracking system, a compact set of metrics will be formed and the prime-notation ($'$) will be used. These metrics will be further developed for the tracking testbed experiment – and new metrics introduced – in Section 4.2.6. Recall that each instance of the parametric test generates at most one target, and the test may result in zero, one, or many tracks, i.e., confirmations followed by deletions. Define the true target presence function as

$$\delta'(k) = \begin{cases} 1 & \text{target present} \\ 0 & \text{target absent} \end{cases} \tag{4.1}$$

for each time k in that instance's full set of times $\mathbf{K}$. Referring to each confirmation/deletion as a distinct track identity, define the function $\mathcal{I}'(k)$ which assigns a positive, natural, track identity ($\mathcal{I}'(k) \in \mathbb{N}_1$) or in the no track case ($\mathcal{I}'(k) = 0$) at

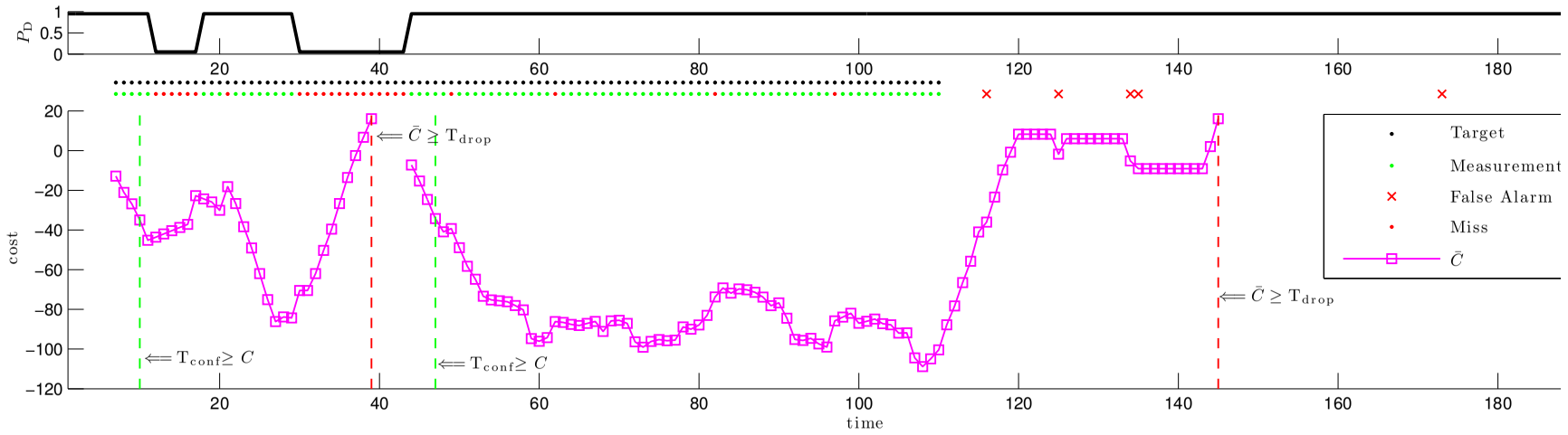


Figure 4.1: An illustration of a single instantiation of the parametric test. Here, a true target arrives at frame 7 and departs at frame 110 (black dots). Two simulated occlusions have occurred: one begins at frame 12, and the other at frame 30. The top plot of P_D illustrates the occlusion events. The magenta line of connected boxes represents the track cost. A track is confirmed on the target at frame 10 (dashed green line), and deleted at frame 39. A subsequent track is confirmed on the target at frame 47, and deleted at frame 145. Notably, this track persisted 35 frames beyond the target's departure due to four false alarms (red x's). This instantiation used uniform background statistics rather than CAT.

each time k. For convenience, require that $\mathcal{I}'(k') = 1$ when the system confirms the first track at some time $k' > 0$, and the track identity is incremented by 1 for any subsequent track confirmation. Track completeness is defined as

$$\bar{\mathcal{M}}'_{\text{comp}} = \frac{|k : \mathcal{I}'(k) \neq 0 \bigcap k : \delta'(k) = 1|}{|k : \delta'(k) = 1|} \,, \tag{4.2}$$

where $|\cdot|$ is the set counting operator. Thus, completeness refers to the ratio of the time that the target is present *and* covered by a confirmed track, to the time in which it is present. Completeness can naturally be extended to a collection of many instances – as in a Monte Carlo experiment. It lies on the range $[0, 1]$, where 1 indicates ideal coverage. Specific track identity is of no consequence to $\bar{\mathcal{M}}'_{\text{comp}}$. Should track deletion occur followed immediately by confirmation of a new track, i.e., an identity-swap, $\bar{\mathcal{M}}'_{\text{comp}}$ is not penalized.

Conversely, track purity is not concerned with coverage, but with track identity over the entire scenario $\mathbf{K}$:

$$\bar{\mathcal{M}}'_{\text{pure}} = \frac{|k : \mathcal{I}'(k) = \text{mode}\,\mathcal{I}'(\mathbf{K})|}{|k : \mathcal{I}'(k) \neq 0|}, \tag{4.3}$$

which is the ratio of the times in which the target is assigned its most frequently occurring identity to the frames in which it is assigned any track identity. Again, purity is extensible to Monte Carlo analysis. It lies on the range $(0, 1]$, where 1 indicates that when identity assignment occurs, it remains entirely consistent. As $\bar{\mathcal{M}}'_{\text{pure}} \to 0$, identity swapping occurs more frequently.

A final metric is introduced which is concerned with the system's ability to estimate the presence or absence of targets. In this parametric experiment, at any time k the system will either have, or not have a confirmed and not-yet-deleted track, i.e., an active track. Thus, the cardinality of the system's tracks is 1 or 0. Also at any time k, a target will either be present or absent; the cardinality of the truth is 1

or 0. The difference in cardinality is the delta-cardinality, and is

$$\Delta'_{\text{card}}(k) = \begin{cases} 0 & \text{target present, active track} \\ 1 & \text{target absent, active track} \\ 1 & \text{target present, no active track} \\ 0 & \text{target absent, no active track} \end{cases} \tag{4.4}$$

at each time k. The mean delta-cardinality of an instance – or a Monte Carlo set of many instances – is

$$\bar{\Delta}'_{\text{card}} = \frac{\sum_{k \in \mathbf{K}} \Delta'_{\text{card}}(k)}{|\mathbf{K}|} \ . \tag{4.5}$$

The delta-cardinality is ideally 0. In this parametric experiment, the upper-bound is 1, e.g., the system always has an active track when the target is absent, and never has an active track when the target is present. Note that delta-cardinality is related to completeness, except that delta-cardinality penalizes the system when the target is absent and an active track remains.

4.1.2 The Time-Domain Nature of Occlusions. Subjective analysis of prior tracking systems within challenging environments suggests that occlusions are a leading contributor to tracking failures. A first-order characterization of occlusions in tracking scenarios is given by occlusion arrival rate and occlusion duration. Here, the occlusion arrival rate is drawn from a geometric distribution defined by a certain per-frame probability of a new occlusion; occlusion duration is drawn from a uniform distribution between a minimum and maximum duration. A test matrix has been formed in which these two aspects of occlusions are swept through typical ranges. The occlusion arrival element is represented by the `pOcclusion` parameter. This is treated as the per-frame probability of a new occlusion beginning, conditioned on the simulation being in an occlusion-free state during the previous frame. Occlusions arrive according to a Bernoulli trial with the `pOcclusion` probability, such that the inter-occlusion times take on the geometric distribution. This behavior was selected

to mimic real-world occlusions, which are generally independent and discrete events caused by background structure. Notably, any number of occlusions are permitted in a simulation run. As the occlusion duration is drawn from a uniform distribution, there are two underlying parameters – the minimum and maximum bounds. Generally these are set relatively far apart to allow for both brief and extended occlusions. In this experiment, they are combined into a single quantized parameter where the minimum is only slightly smaller than the maximum. This was done in order to increase the granularity of the results.

The resulting test serves as a projection of the overall operating-condition space onto a relatively simple two-dimensional parameter field. This field can be related to real-world tracking scenarios, as illustrated in Figure 4.2. Consider the case where `pOcclusion`$= 0.01$ and `minOcclusionDur,maxOcclusionDur`$= 5, 10$. Further assuming a real-world sensor with a 10Hz frame-rate, this yields an expected inter-occlusion time of 10 seconds and occlusions between $0.5s$ and $1.5s$ in duration. This is notionally equivalent to a benign, rural tracking scenario in which a target at $20\frac{m}{s}$travels a route with occluding background elements, e.g., trees, spaced $200m$ apart and spanning $10m$ to $30m$ in width. A more challenging case exists when `pOcclusion`$= 0.1$ and `minOcclusionDur,maxOcclusionDur`$= 20, 25$. This corresponds to an urban-canyon tracking scenario in which a $10\frac{m}{s}$target is generally visible for only $10m$ and occluded for 20-25m. Also notionally illustrated in Figure 4.2, there is reason to expect a nonlinear relationship between occlusion duration/frequency and tracking difficulty. If the percentage of time in which a target is occluded can be used as to estimate tracking difficulty, then clearly the effects of occlusion duration and frequency are multiplicative, rather than cumulative in nature.

4.1.3 Results. A Monte Carlo analysis has been performed with 200 runs in each cell of the two-dimensional parameter space. The resulting metrics for this test appear in Figure 4.3 for the uniform case. The test has been repeated with CAT enabled, resulting in the metrics of Figure 4.4.

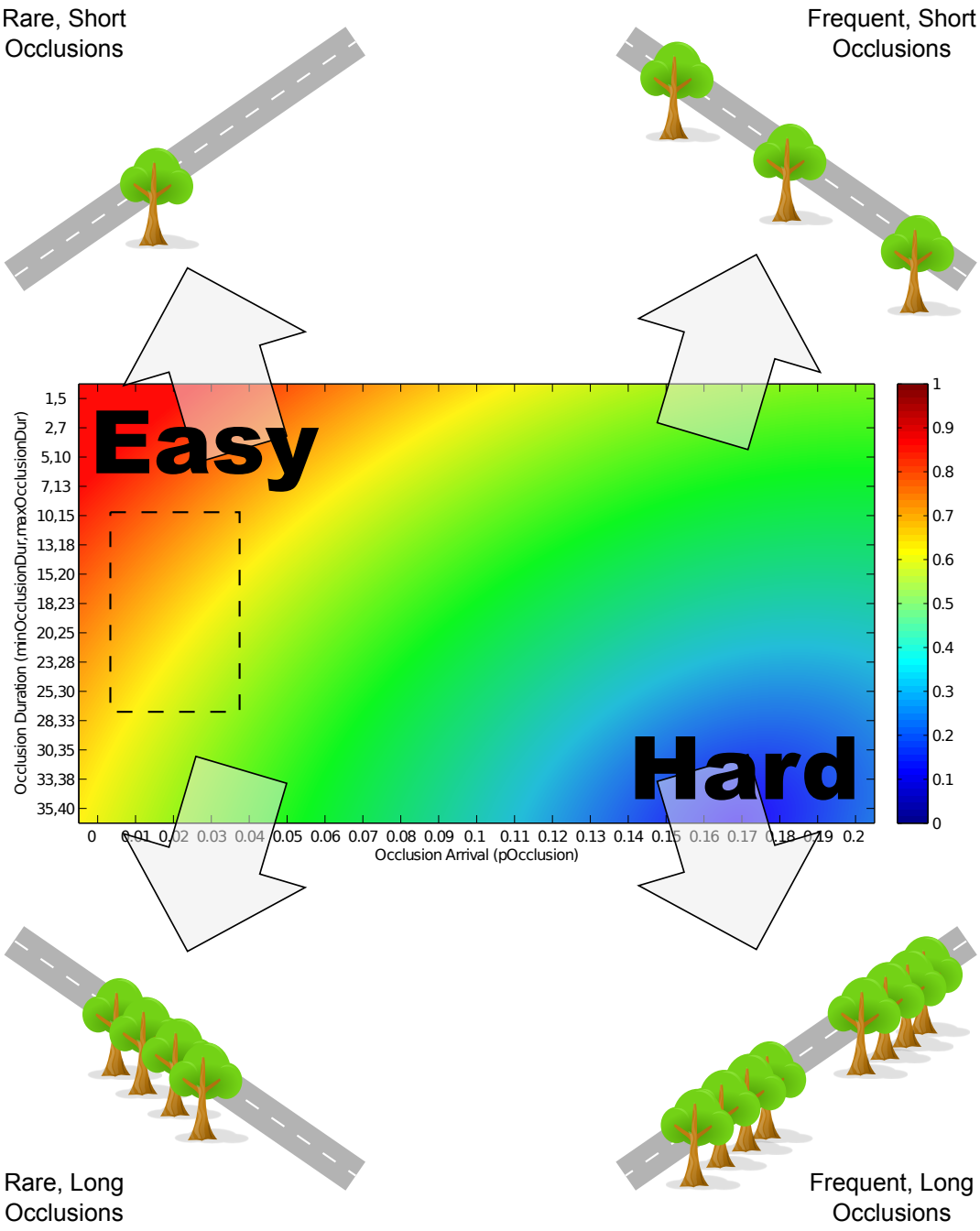
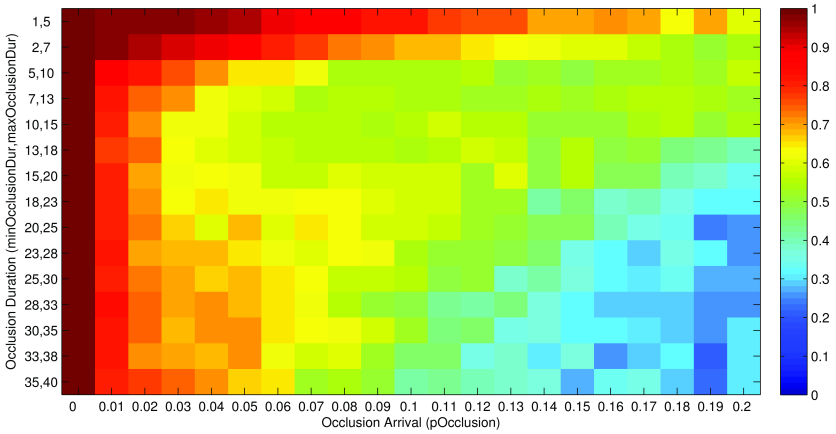
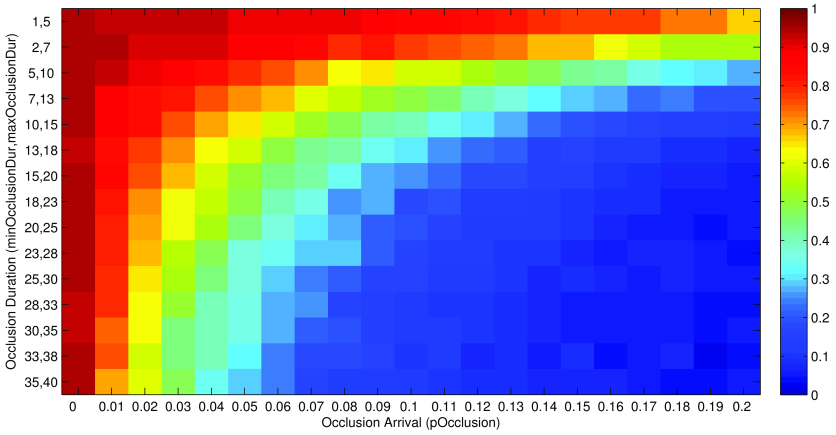


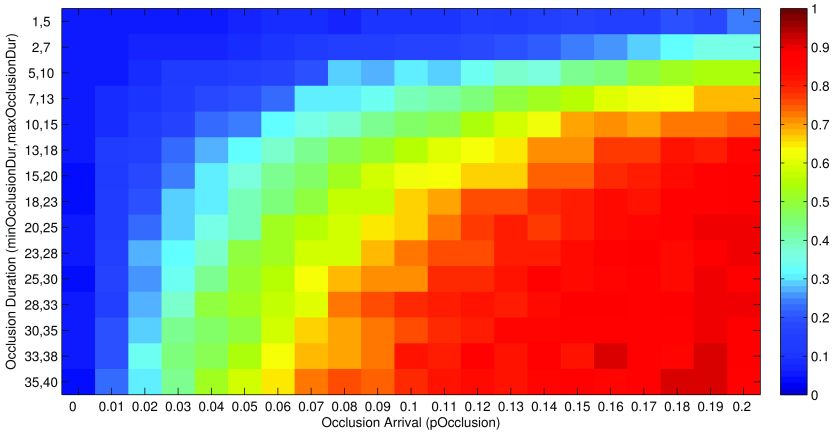
Figure 4.2: Notional meaning of the time-domain nature of occlusions. The operation-condition space has many dimensions; here it has been collapsed into two dimensions that describe the first-order effect of occlusion. The x-axis represents the arrival rate of occlusions. The y-axis represents the duration of occlusions. The colorbar indicates a subjective assignment of difficulty. The shape of the difficulty field is intended to emphasize the nonlinear nature of the OC space. Simple examples are given for the four corners of the space. For reference, the dashed box represents the portion of the space in which the majority of the synthetic tracking data – described in Section 4.2 – lies.



(a) Purity

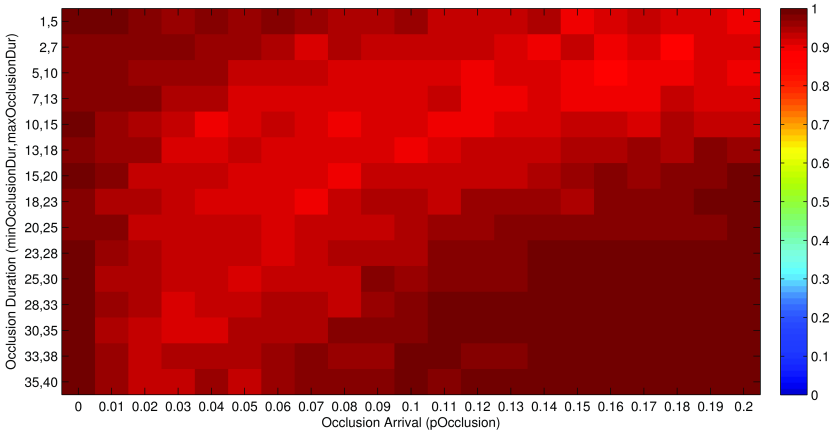


(b) Completeness

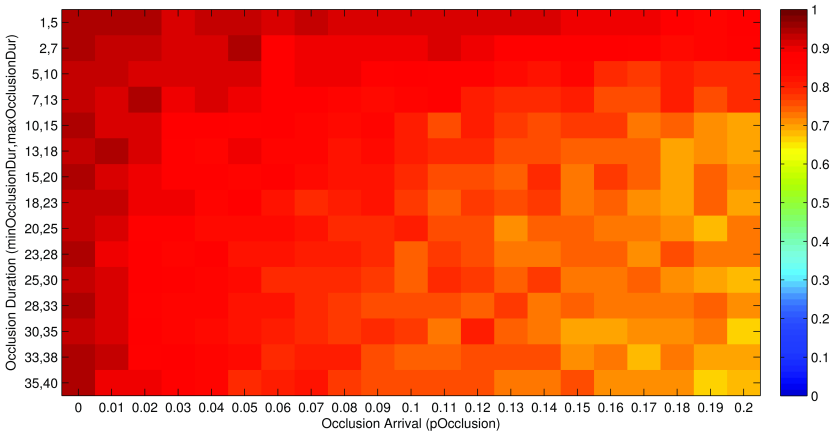


(c) Delta cardinality

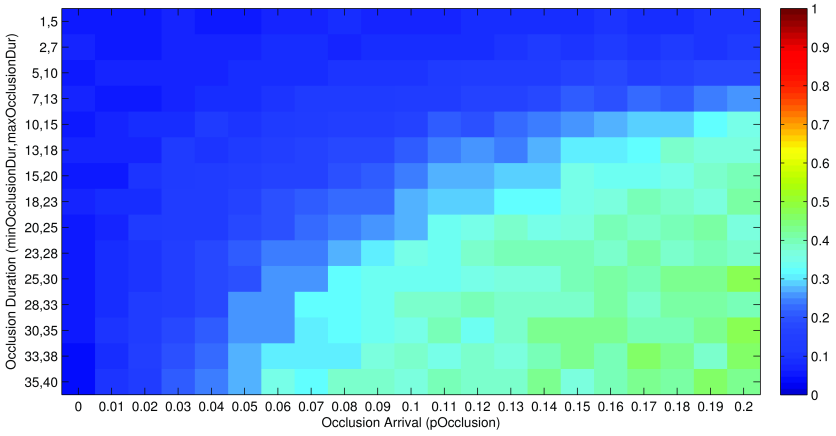
Figure 4.3: Parametric experiment results for the time-domain nature of occlusions. In this test, uniform background statistics are utilized. The three metrics of (a) purity, (b) completeness, and (c) delta cardinality are plotted within two extrinsic dimensions of the OC space: occlusion arrival (`pOcclusion`), and occlusion duration (`minOcclusionDur,maxOcclusionDur`).



(a) Purity



(b) Completeness



(c) Delta cardinality

Figure 4.4: Parametric experiment results for the time-domain nature of occlusions. In this test, CAT statistics are utilized. The three metrics of (a) purity, (b) completeness, and (c) delta cardinality are plotted within two extrinsic dimensions of the OC space: occlusion arrival (`pOcclusion`), and occlusion duration (`minOcclusionDur`,`maxOcclusionDur`).

A subjective analysis of the uniform results in Figure 4.3 agrees with common sense: the low `pOcclusion`, low `minOcclusionDur,maxOcclusionDur` quadrant of the field is the least challenging and yields the best performance – i.e., high purity and completeness with low delta-cardinality. The high `pOcclusion`, high `minOcclusionDur,maxOcclusionDur` quadrant of the field is the most challenging and yields the lowest performance – i.e., low purity and completeness with high delta-cardinality. Furthermore, a curved shape exists within the field for all three metrics. This indicates a nonlinear, worse-than-sum effect between these dimensions. In marked contrast, the same three metrics for the CAT test in Figure 4.4 show a significant increase in purity and completeness, and decrease in delta cardinality. This represents the performance gain of CAT relative to the baseline uniform background statistics. Several salient observations regarding the results follow:

- At the far left side of the field where occlusions never occur (`pOcclusion`$\rightarrow 0$), CAT is of no use. However, there is no statistically significant evidence of CAT causing harm in this case.
- At the center of the field (`pOcclusion`$= 0.1$ and `minOcclusionDur, maxOcclusionDur` $= 18{,}23$), CAT improves purity by 50% ($0.6 \rightarrow 0.9$), completeness by 400% ($0.2 \rightarrow 0.8$), and delta cardinality by 50% ($0.6 \rightarrow 0.3$). This is a very significant performance improvement in all metrics. This point in the field represents a very challenging tracking scenario. In the tracking experiment in Section 4.2, extreme portions of the vignette with clustered tree canopies will approach this level of occlusion challenge.
- An aggregated region of the middle of the lefthand side of the field – akin to the dotted box in Figure 4.2 – represents the range of the occlusion difficulty which will be tested in the tracking experiment in Section 4.2. Notably, this spans the region from where the parametric test shows little performance gain with CAT to the region where CAT creates a significant improvement.

With the usefulness of CAT now demonstrated across a relevant OC space, an experiment is designed to test the sensitivity of CAT to intrinsic tuning parameters. The key intrinsic parameter in CAT is $P_{\mathrm{D}}^{\mathrm{CAT}}$, the modeled probability of detection. In the parametric experiment, $P_{\mathrm{D}}^{\mathrm{CAT}}$ always takes one of two values. When no occlusion is occurring it is (`CAT_Pd_clear`), which is a model of the simulation's true probability of detection outside of occlusion (`nominalClearPdTrue`). When an occlusion is underway it is (`CAT_Pd_occluded`), which is a model of the simulation's true probability of detection during occlusion (`nominalOccludedPdTrue`). This second value is anticipated to be the most likely source of tuning sensitivity; it impacts the system when occlusions are in force and CAT is most needed. To cast the importance of these parameters on the CAT background statistics of Section 3.4.4, the following observations are made:

- The case where `CAT_Pd_occluded` $\approx$ `nominalOccludedPdTrue` corresponds to success of the background modeling stage in determining how severe an occlusion might be.
- The case where `CAT_Pd_occluded` $\ll$ `nominalOccludedPdTrue` corresponds to a region where the background modeling declared an occlusion to be more severe than it truly is.
- The case where `CAT_Pd_occluded` $\gg$ `nominalOccludedPdTrue` corresponds to a region where the background modeling failed to detect a true occlusion, or estimated it to be less severe than it truly is. This is a particularly troublesome case, as there is a strong likelihood for missed detections during the poorly modeled occlusion; the tracker is more apt to erroneously delete the track when this happens.

The sensitivity experiment of the parametric system is a set of Monte Carlo trials across a two dimensional parameter space. In this space, the two dimensions of `CAT_Pd_occluded` and `nominalOccludedPdTrue` are each swept across the range $[0, 0.5]$. Each cell in this space corresponds to 200 Monte Carlo trials. The results

of the experiment are illustrated in Figure 4.5. If the CAT technique had proven to be extremely sensitive to a $P_{\mathrm{D}}^{\mathrm{CAT}}$ mismatch, one would expect the results to show good performance on the diagonal, i.e., `CAT_Pd_occluded` = `nominalOccludedPdTrue`. However, this is not apparently the case. Performance is relatively uniform across the majority of the space, except for the lower left corner, which corresponds to

$$\texttt{CAT\_Pd\_occluded} \gg \texttt{nominalOccludedPdTrue} \bigcap \texttt{CAT\_Pd\_occluded} > 0.3 \quad .$$

The first term of the intersection corresponds to the third observation above, in which the model has grossly underestimated a true occlusion and performance is expected to suffer. The significant change in the metrics near `CAT_Pd_occluded`= 0.3, however, suggests that some intrinsic threshold of the track maintenance logic is being crossed. An examination of the track-drop threshold $\mathrm{T}_{\mathrm{drop}}$ in Equation (3.17) and the track deletion logic in Equation (3.18) is helpful to understand this effect. When `nominalOccludedPdTrue`$\rightarrow 0$, the likelihood of updating the track with a measurement is low. Thus, the windowed cost $\bar{C}(k) \rightarrow \mathrm{N}_{\mathrm{drop}} \ln[1 - P_{\mathrm{D}}]$. In order for a track deletion to occur in this case,

$$\lim_{\texttt{nominalOccludedPdTrue}\rightarrow 0} \left[\bar{C}(k) \geq \mathrm{T}_{\mathrm{drop}}\right] \quad . \tag{4.6}$$

Incorporating Equation (3.17) and simplifying, this becomes

$$\begin{aligned} \mathrm{N}_{\mathrm{drop}} \ln[1 - P_{\mathrm{D}}] &\geq -\mathrm{M}_{\mathrm{drop}} \ln\left[1 - P_{\mathrm{D}}\right] - (\mathrm{N}_{\mathrm{drop}} - \mathrm{M}_{\mathrm{drop}}) \ln\left[\frac{P_{\mathrm{D}} p_{\mathrm{drop}}^{\mathrm{SS}}(z|x)}{\beta_{\mathrm{FA}}}\right] \\ \ln\left[1 - P_{\mathrm{D}}\right] &\leq \ln\left[\frac{P_{\mathrm{D}} p_{\mathrm{drop}}^{\mathrm{SS}}(z|x)}{\beta_{\mathrm{FA}}}\right] \\ P_{\mathrm{D}} &\geq \left[\frac{p_{\mathrm{drop}}^{\mathrm{SS}}(z|x)}{\beta_{\mathrm{FA}}} + 1\right]^{-1} \quad . \end{aligned} \tag{4.7}$$

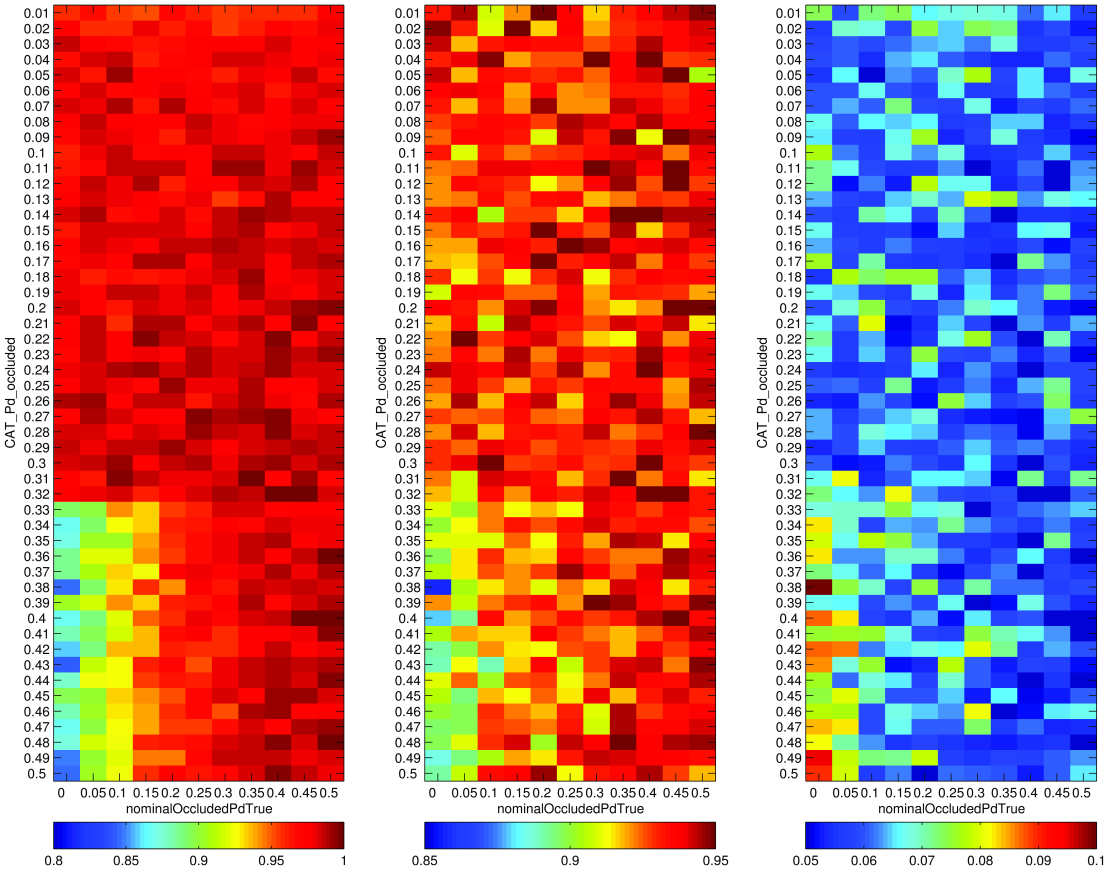
So, the ability to delete any track as `nominalOccludedPdTrue`$\rightarrow 0$ is dominated by an inequality of three intrinsic tuning parameters, i.e., constants. In this particular

experiment, the inequality becomes $P_{\mathrm{D}} \geq 0.3$, explaining this significant change in the results. In Figure 4.5, the top region of the plots represents where the inequality is violated and tracks cannot be deleted during occlusion. The favorable performance in this region is largely a consequence of the simplifying single-target assumptions in the parametric experiment framework. Nevertheless, Equation (4.7) is an important consideration in a CAT system. Beyond this finding, the original intent of sensitivity analysis seems to be favorably answered. In the region satisfying Equation (4.7) – the bottom portion of the plots in Figure 4.5 – the system performance does not depend on carefully matching `CAT_Pd_occluded` to `nominalOccludedPdTrue`.

4.2 Tracking Experiment

While the parametric experiment was extremely useful in analyzing the effects of various operating conditions (OC's), further study was required that incorporated the full aspects of the MHT system. Furthermore, a full rendering of HSI data was required to allow incorporation of the detection algorithms and testing of the SRM methods presented in this section. As such, fewer instances over a variety of OC's is available, but the fidelity of the simulation and ensuing results is compelling and in fact necessary to validate the parametric study.

The scenario has been rendered with DIRSIG at 10Hz temporal sampling with a notional hyperspectral instrument mounted to an airborne platform and oriented towards nadir. Platform motion has been excluded for simplicity, but is more generally resolved with registration techniques. The observation geometry and optical design yield a ground-sample-distance of 0.5m and an overall field-of-view of 0.3km^2. A spectral bandwidth of $0.4\mu - 1\mu$ at 0.01μ resolution results in 61 bands. This is representative of a realizable silicon-based visible-light MOS instrument. The HSI data have been rendered with 560 lines and 880 samples of spatial resolution. Each sample is sub-sampled in a 3x3 fashion, such that nine independent spectral radiance values are computed and linearly mixed. This approximates the spectral mixing which is common to real HSI data.



(a) Purity (b) Completeness (c) Delta cardinality

Figure 4.5: Parametric experiment results for the time-domain nature of occlusions. In this test, CAT statistics are utilized. The three metrics of (a) purity, (b) completeness, and (c) delta cardinality are plotted across two related parameters. `CAT_Pd_occluded` is an intrinsic tuning parameter which is enforced during occlusion events, and is intended to estimate the extrinsic parameter `nominalOccludedPdTrue`. In order to visualize the relatively narrow range of results for these three metrics in this Monte Carlo experiment, the range of the colormap is not $[0, 1]$ in these plots.

Approximately 1000 frames of imagery have been rendered, accounting for 100 seconds of moving-vehicle data. The dataset has been rendered in dense-hyperspectral mode. This provides more hyperspectral pixels than would be available from any MOS instrument, and therefore is down-sampled according to the exploitation algorithm and SRM under test. Additionally, panchromatic imagery has been derived from the spectral data and represents the video-rate imaging channel available on the MOS instrument. A traditional frame-to-frame motion detection technique has been performed on this panchromatic channel, resulting in motion detections suitable for tracking. The intentional addition of modelled noise into the data results in false-alarm motion detections. Occlusion, illumination effects, and low-contrast vehicles result in frequent missed-detections. Two groups of tracking experiments have been performed: a series of uniform-statistic "control" tests and an innovative context-aided test. The vehicle population and motion are the same for both tests. The motion detection process was precomputed and stored such that all tests would encounter exactly the same false-alarms and missed-detections.

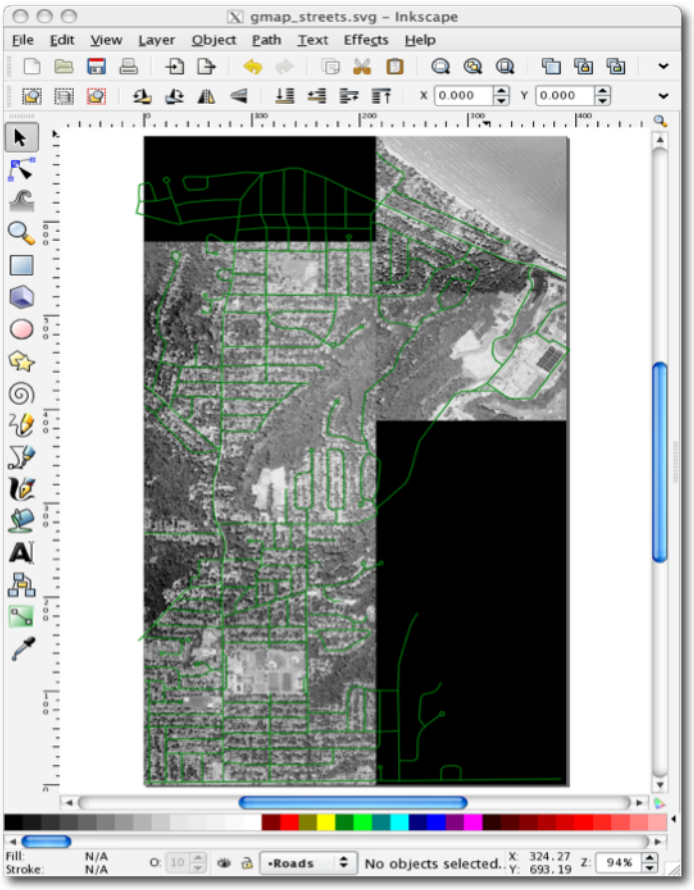
For the uniform-statistic tests, values for P_{D}, β_{NT}, and β_{FA} were set according to the final row in Table 3.1. These values were empirically determined and are known to produce good tracking results for this combination of scenario, motion-detector, and MHT tracker. For the context-aided test, a full hyperspectral cube was formed from the data. The manual, offline, background modeling technique described in Section 3.4.1 was applied, resulting in a functional classification of background materials in the scene. These were converted into spatially-dependent background statistics maps according to the entries in Table 3.1.

4.2.1 Data Synthesis. The tracking experiment has been formed around the RIT "Megascene" scenario developed by RIT and distributed with DIRSIG. Megascene is a geo-specific model of a portion of Rochester, New York. Suburban in nature, if offers a moderately dense road network and many occlusions, some of them short in duration, while others are long in duration. An area of approximately

Figure 4.6: A pseudocolor rendering of the Megascene synthetic environment. This image was derived from a hyperspectral rendering performed with DIRSIG.

$0.3km^2$ near Dake Junior High School $(43.217851N, 77.598447E)$ serves as the area of interest, shown in Figure 4.6.

A population of 211 moving vehicles has been formed, and random waypoint navigation has resulted in a difficult tracking challenge. The Simulation of Urban Mobility (SUMO) traffic simulation package [33] has been used to apply accurate kinematics and traffic rules to the vehicle motion. SUMO is a microscopic (discrete vehicles), discrete-time dynamics tool with an emphasis on traffic behavior for large and congested road networks, illustrated in Figure 4.7. The vehicle geometries consisted of several generic sedan, station-wagon, and pickup truck models available in the Megascene distribution. The vehicles were "painted" with hyperspectral reflectance signatures provide by the Air Force Research Lab in a 41 vehicle dataset. These signatures were measured from donor civilian vehicles by an Advanced Spectral Devices Incorporated Field Spectrometer and are shown in Figure 4.8. For each of the 211



(a) SUMO road network

(b) SUMO simulation dynamics

Figure 4.7: An illustration of the SUMO road traffic simulator. In (a), a vector representation of the road network of the Rochester, New York area is shown. This road network became the input to SUMO. In (b), a screenshot is given of SUMO processing a random traffic simulation for incorporation in the Megascene environment.

vehicle trajectories output by SUMO, a vehicle geometry and hyperspectral signature were randomly drawn and assigned for the duration of the experiment.

4.2.2 Semi-Automated Background Modeling. The first stage of the tracking experiment required an implementation of the semi-automated background modeling technique described in Section 3.4.1. The empirically determined NDVI threshold was set at 0.18, resulting in the vegetation detection illustrated in Figure 4.9. A comparison to the pixel-level-truth output from DIRSIG indicates a probability of vegetation detection of 0.93, where the criterion for success was that pixels with a true majority of vegetation content should be declared vegetation, and others should not. The proportion of declared vegetation pixels which were not a true majority of vegetation is 0.003. Next, the empirically determined tree-index threshold was set at 1×10^{-5}, resulting in the tree canopy detection illustrated in Figure 4.10. Again, a comparison to the pixel-level-truth indicated a probability of tree canopy detection of 0.80, and a proportion of false detection of 0.086. Notably, this detection probability is computed against the entire population of pixels in the image. If the detection probability is considered against only the population of thresholded vegetation pixels, it rises somewhat to 0.89. The next portion of the background modeling consisted of selecting training samples for the AGRLVQI classifier. As the vegetation portions of the scene have already been identified, this step is primarily concerned with discerning the remaining functional elements of the scene, e.g., building materials and pavement materials. Figure 4.11 illustrates portions of the image from which training samples were manually selected. Experience suggests that separating surface pavement materials from common roofing materials is a challenging problem, particularly since many of the constituent elements are the same, e.g., asphalt, gravel, and sand. An effective means to overcome this problem is to separate the two functional classes into several sub-classes, where each sub-class is somewhat spectrally distinct. Asphalt impregnated roofing shingles, for example, tend to include dyes for aesthetic purposes. These dyes are dominated by several common colors. Surface pavement tends to be

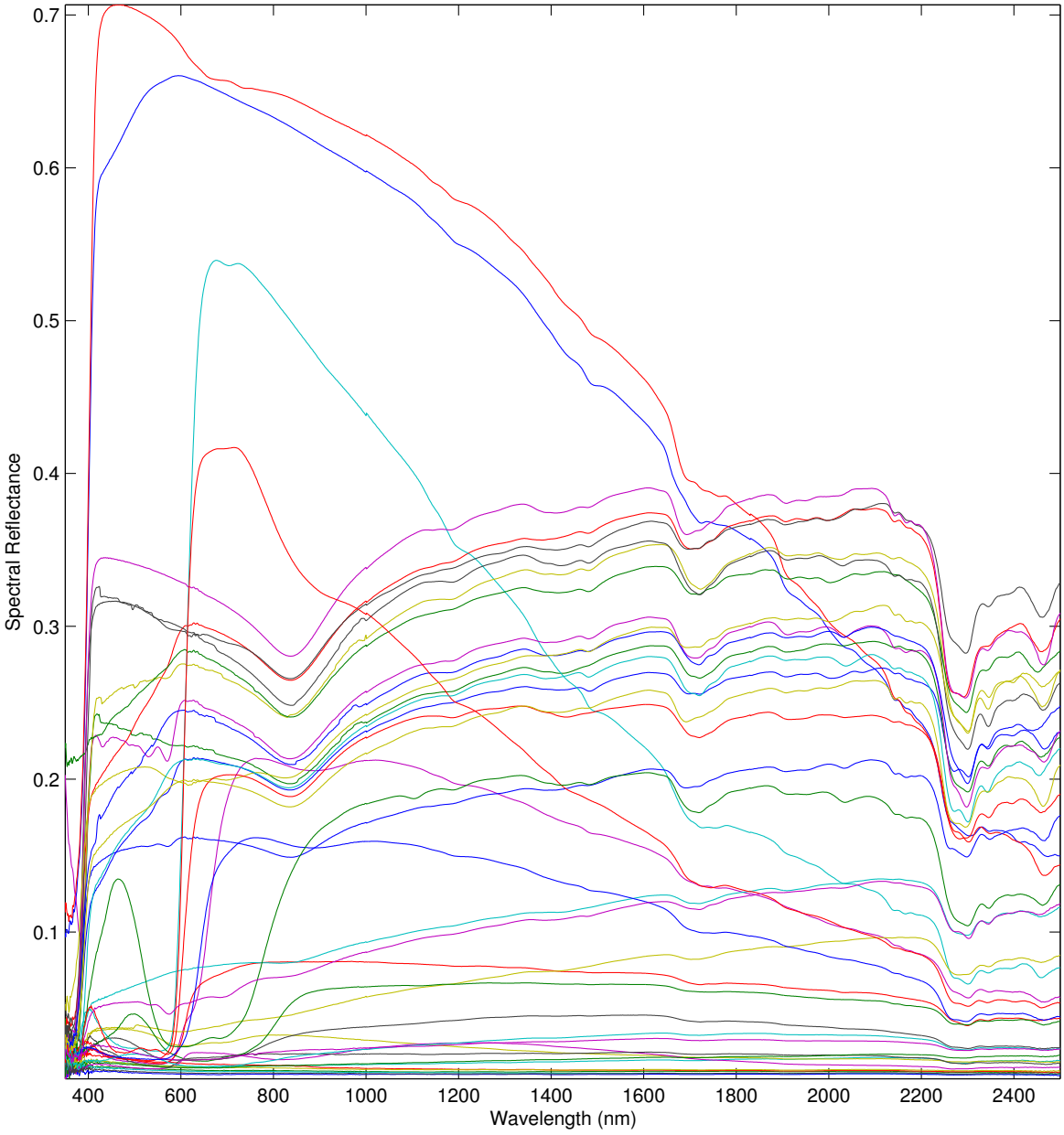


Figure 4.8: A plot of the mean of the hyperspectral reflectance signatures for 41 vehicles measured by AFRL. A variety of makes, models, years, and colors were included.

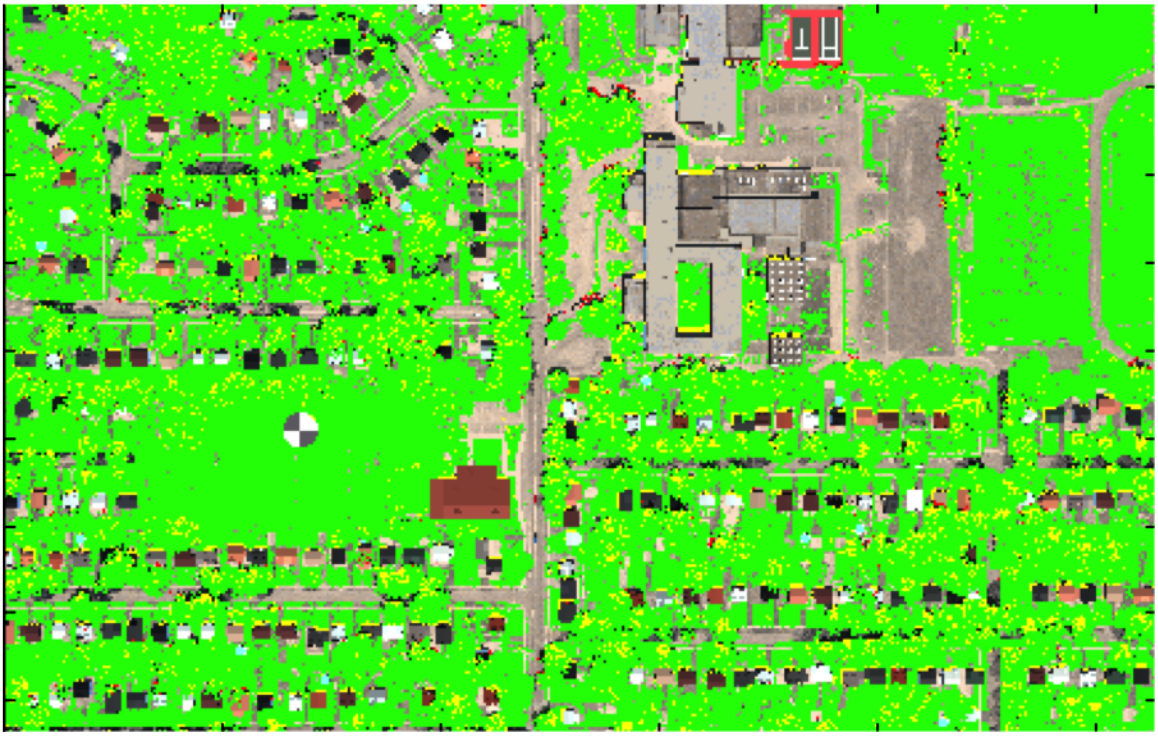


Figure 4.9: An illustration of the vegetation detection stage of background modeling. The underlying image is a pseudocolor rendering of the hyperspectral cube. The bright green and yellow colors indicate pixels which exceeded the thresholded NDVI test and were declared vegetation. According to truth, the bright green pixels are majority vegetation, and therefore represent accurate detections. The bright yellow pixels are not majority vegetation, and therefore represent false detections.

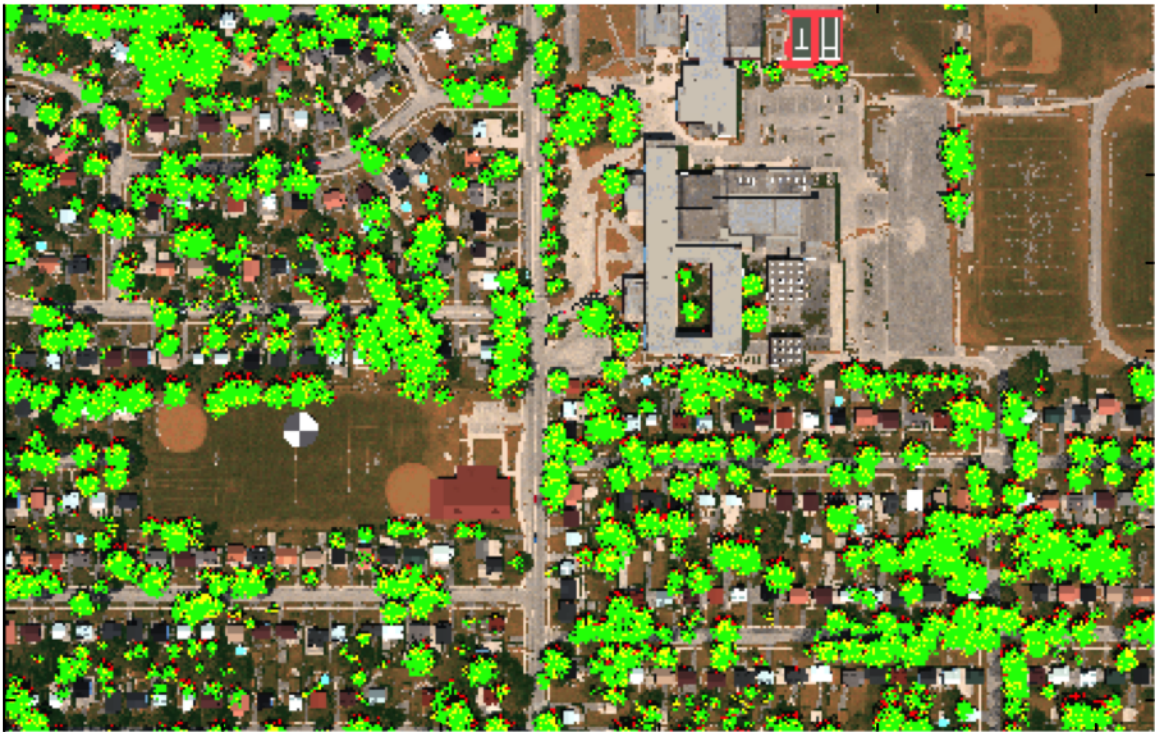


Figure 4.10: An illustration of the tree canopy detection stage of background modeling. The underlying image is a pseudocolor rendering of the hyperspectral cube. The bright green and yellow colors indicate pixels which exceeded the thresholded tree-index test and were declared tree canopy. According to truth, the bright green pixels are majority tree canopy, and therefore represent accurate detections. The bright yellow pixels are not majority tree canopy, and therefore represent false detections.

Table 4.3: The hierarchy of training classes for the AGRLVQI classifier used in the semi-automated background modeling stage.

Training Class	**Functional Material**
light asphalt road	surface pavement
medium asphalt road	
driveway	
parking lot	
water (in pools)	water
pale-blue roof	building materials
white roof	
dark-red roof	
light-red roof	
black roof	
gray roof	
brown roof	
light gravel roof	
dark gravel roof	
medium gravel roof	

of either the asphalt or concrete type, and changes spectrally due to aging and weathering. The hierarchy of training classes selected is given in Table 4.3. The trained AGRLVQI classifier was executed on the non-vegetation portion of the hyperspectral data, resulting in the classification shown in Figure 4.12. Next, the per-band spatial segmentation code is executed on the hyperspectral cube. After collecting regions in which all pixels share a band-wise segmentation solution, a single-band segmentation is achieved. Figure 4.13 illustrates this segmentation. Finally, each region from the spatial segmentation is assigned a functional identity according to the majority vote of its member pixels in the classification results. This results in a labeled map, shown in Figure 4.14 which has lower noise than the original classification results. Values for P_{D} are assigned according to the empirically derived mapping shown in Table 3.1, resulting in the mapped P_{D} shown in Figure 4.15.

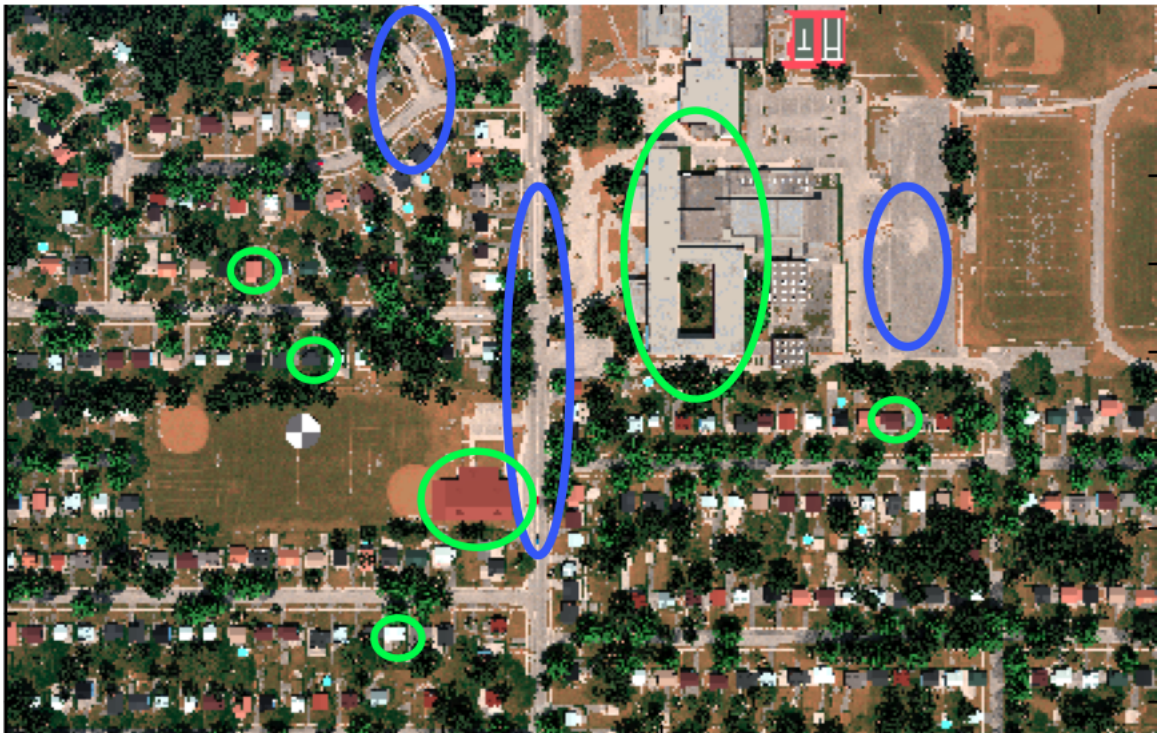


Figure 4.11: An illustration of the AGRLVQI training sample selection stage of background modeling. The underlying image is a pseudocolor rendering of the hyperspectral cube. The blue ellipses indicate areas from which pavement training signatures were taken. The green ellipses indicate areas from which building material training signatures were taken.

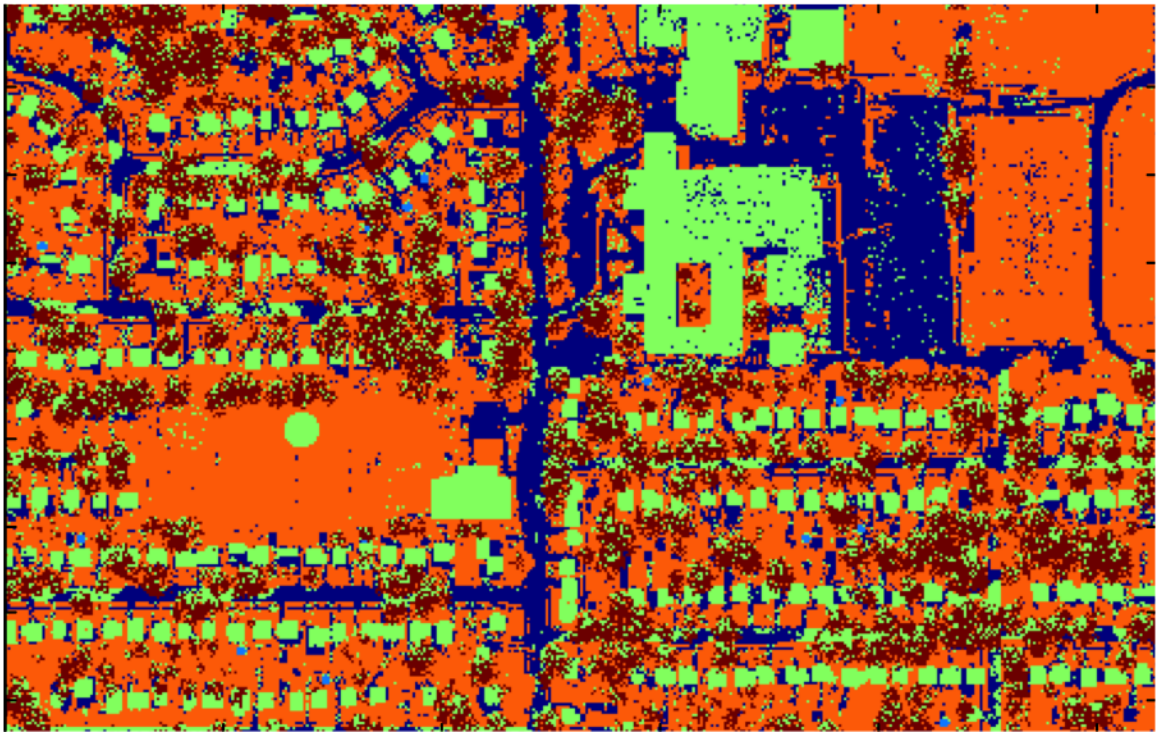


Figure 4.12: An illustration of the AGRLVQI classification results during background modeling. The vegetation pixels were previously identified and excluded from the classification; they are orange in this illustration. The remaining pixels assigned a color according to their parent class (functional material). Blue pixels are road surface, and green pixels are building materials.

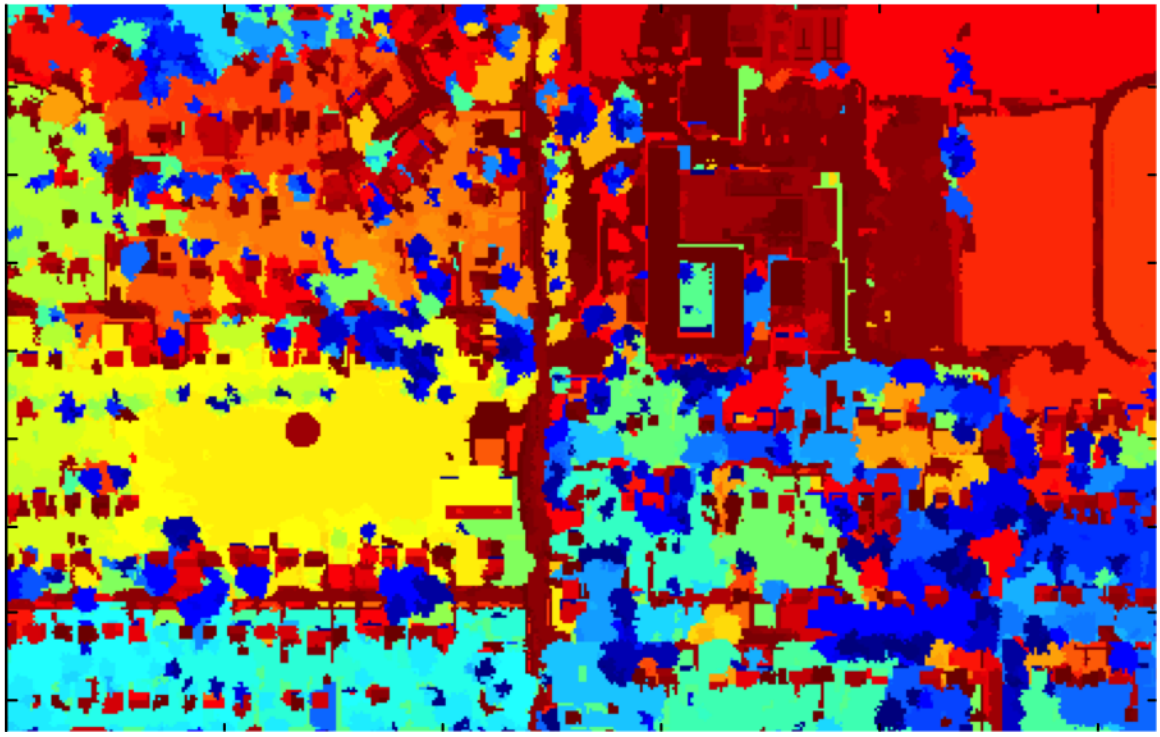


Figure 4.13: An illustration of the spatial segmentation stage of background modeling. Each color represents a region of the scene in which all pixels have a common spatial segmentation throughout all bands of the hyperspectral cube. Notably, the colors are randomly assigned and are allowed to repeat; so non-adjacent regions with similar colors are not necessarily similar.

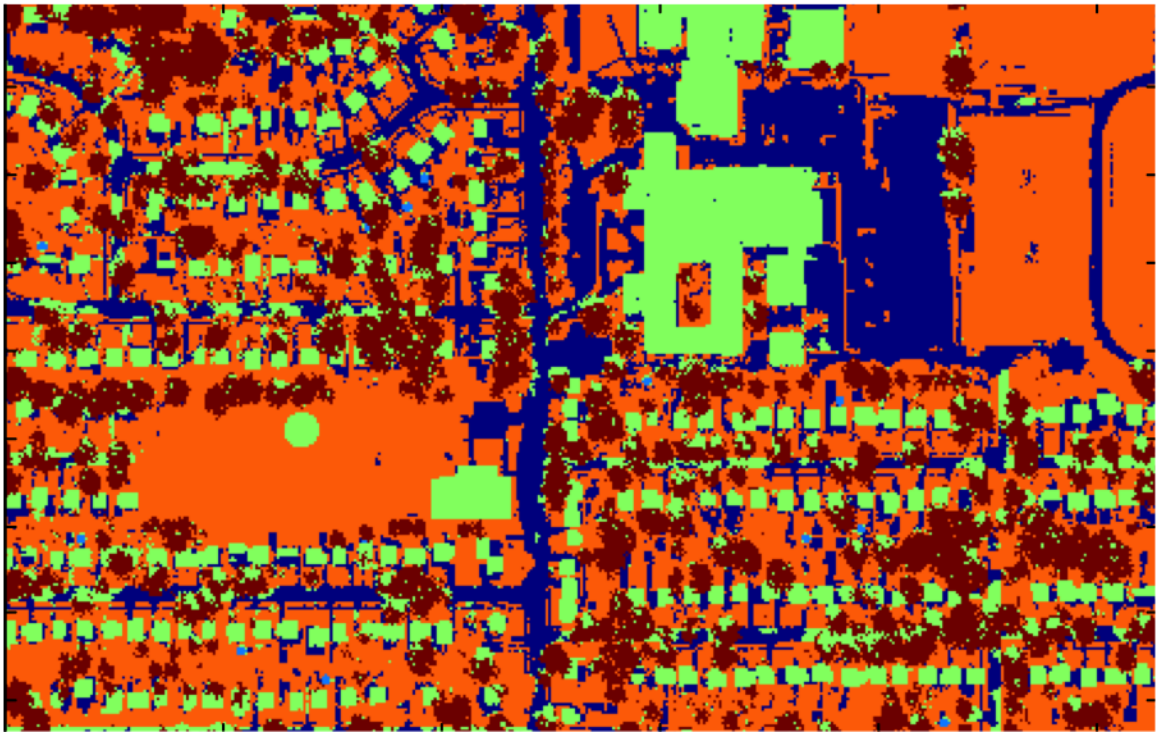


Figure 4.14: An illustration of the final background model for the semi-automated technique. Here, each spatial region is assigned a class label based upon the majority vote of the member pixel's spectral classification results. The color assignment was arbitrary but consistent with Figure 4.12: orange indicates grass, blue indicates pavement, dark red indicates tree canopies, and green indicates buildings.

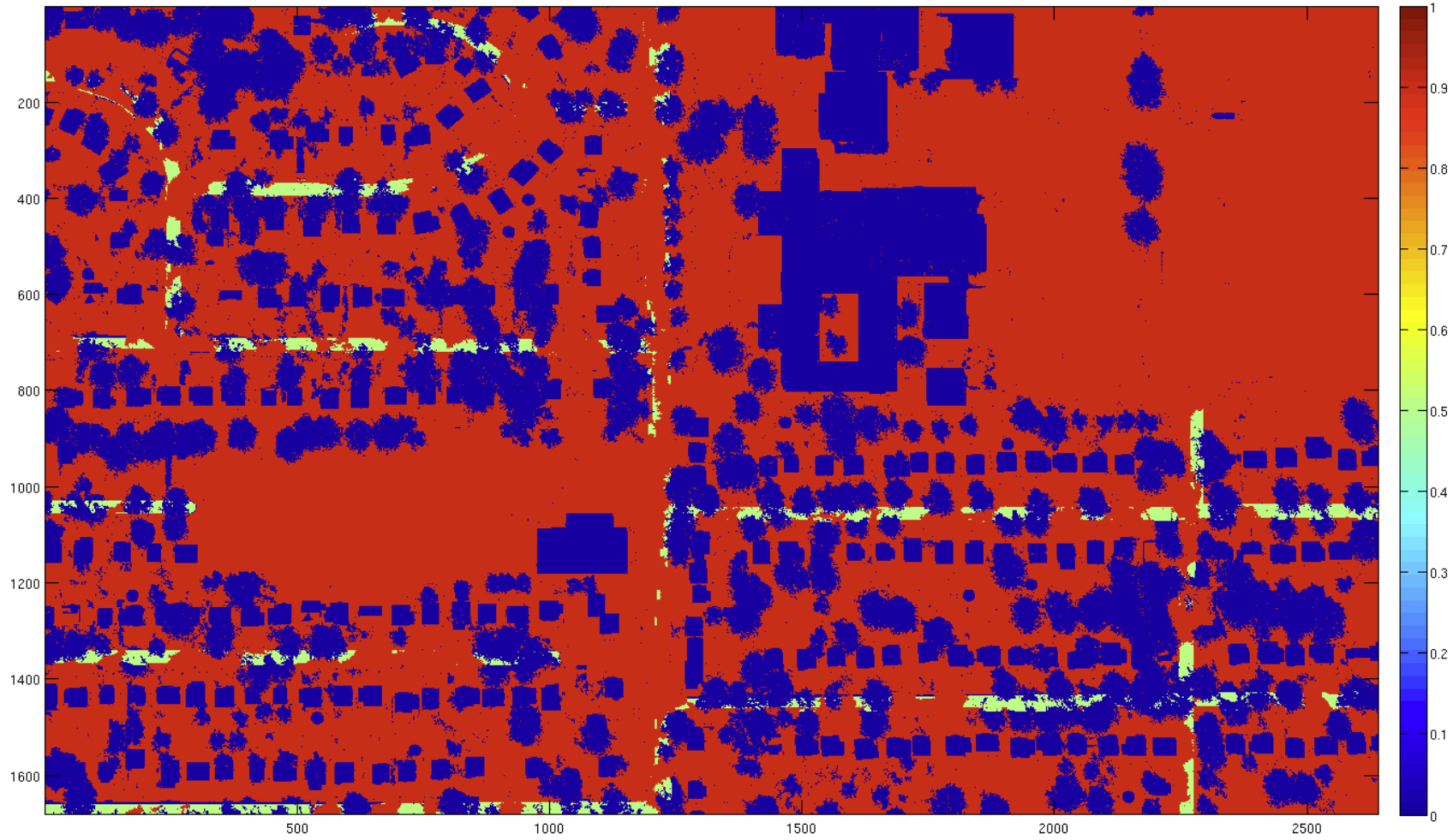


Figure 4.15: An illustration of the mapping of the background model into a spatially dependent P_{D} map. The color bar on the righthand side indicates the values of P_{D} for various colors in the map. The map is dominated by several functional elements: red for open roads and fields, blue for buildings and tree canopies, and green for shadowed roads.

4.2.3 Adaptive Background Modeling. Following the theory presented in Section 3.4.2, an analysis has been performed to simulate the efficiency gain of adaptive background modeling. This process begins with the initialization of a HSI processing system. Recall that any autonomous HSI exploitation technique, which processes both measured (mission-time) HSI data and *a priori* HSI data, must compensate for atmospheric effects. Therefore, an empirical line calibration (ELC) is performed. This experiment utilizes an in-scene spectral calibration fiducial in the form of a large, circular, bicolor panel with known spectral reflectance. The panel is clearly visible in the open field in Figure 4.6. Notably, atmospheric compensation techniques exist which have no dependance upon in scene fiducials; an employment of such a method would be a necessary step towards full automation of this process. The measured spectral radiance of the light and dark portion of the panel is illustrated in Figure 4.16. One sensor acquisition of this panel, coupled with the panel's known reflectance, is sufficient to establish the ELC for the remainder of the experiment. Hence, a radiance-to-reflectance conversion is possible for all subsequent spectral measurements. The spectral region of known-significant water-absorption between 0.93μ and 0.96μ clearly has poor signal to noise ratio and is discarded. The last portion of the HSI processor initialization is concerned with the spectral classifier. The semi-automated background modeling technique for which results were shown in Section 4.2.2 incorporated several indeces and a spectral classifier. In contrast, this adaptive method will only apply an AGRLVQI model which has been trained on reference spectra. Furthermore, the set of classes has been greatly simplified relative to the semi-automated method. Here, the system is permitted no knowledge of scene content with which to "tune" a set of training classes – only opportunistic spectral libraries are used to form a minimal set of functional classes. The training data is sourced from the United States Geological Survey (USGS) [16], the Nonconventional Exploitation Factors Dataset (NEFDS) [4], and the Advanced Spaceborne Thermal Emission Reflection Radiometer (ASTER) spectral library [9]. The hierarchy of training classes to functional materials is given in Table 4.4, and the training classes are illustrated in Figure 4.17. Recalling Equa-

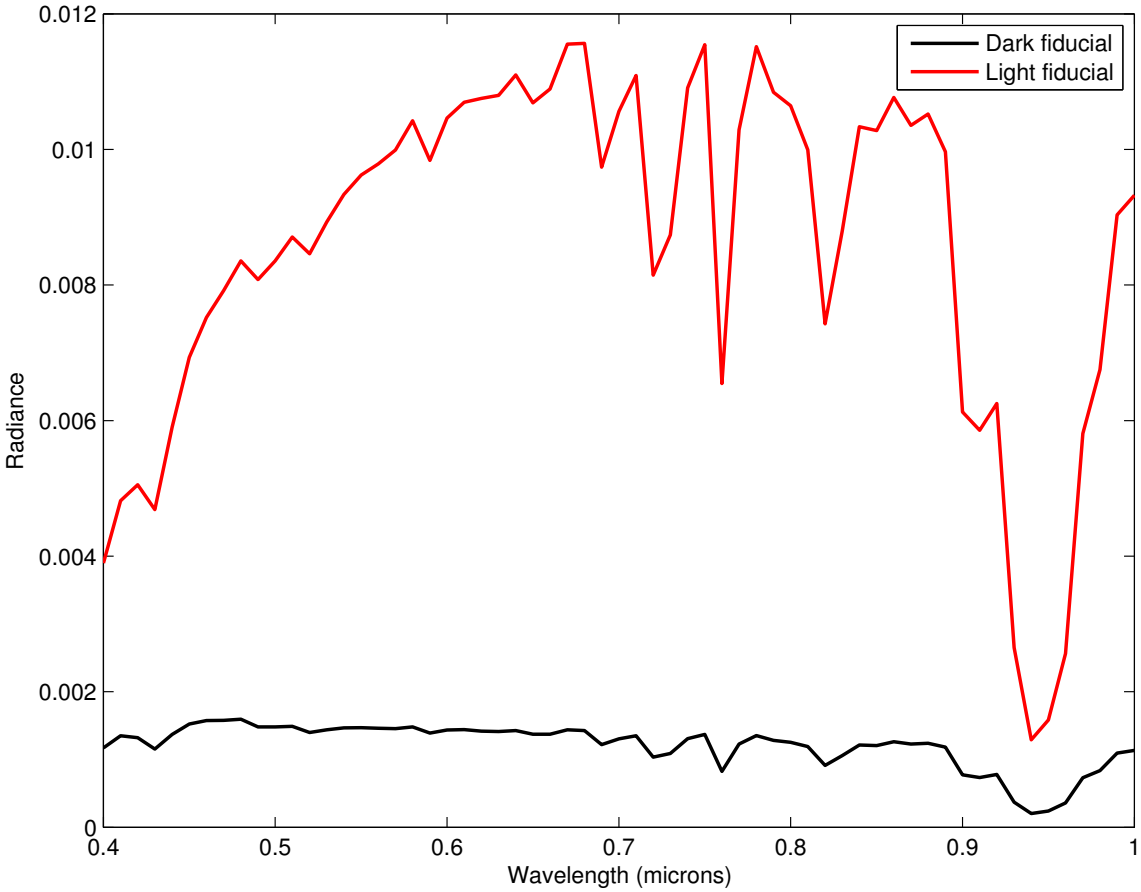


Figure 4.16: The measured radiance of the bicolored spectral calibration fiducial. While both materials were designed to be relatively constant in their spectral reflectance, the atmospheric absorption has caused significant attenuation in portions of the spectrum. The most significant attenuation is centered at 0.94μ, which is a known water absorption band.

Table 4.4: The hierarchy of training classes for the AGRLVQI classifier used in the adaptive background modeling stage.

Training Class	**Functional Material**
asphalt pavement concrete pavement	pavement
asphalt roofing shingles	roof
distressed/healthy grasses soils	grass

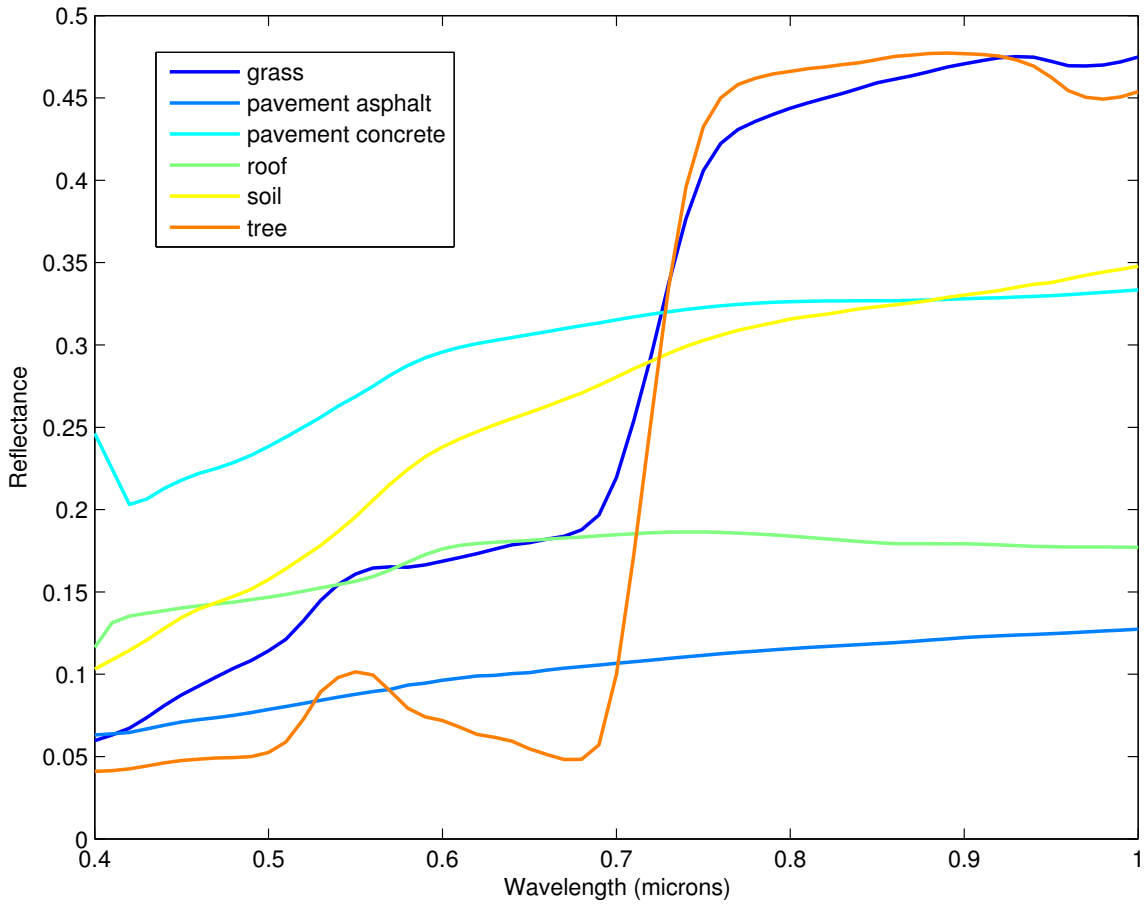
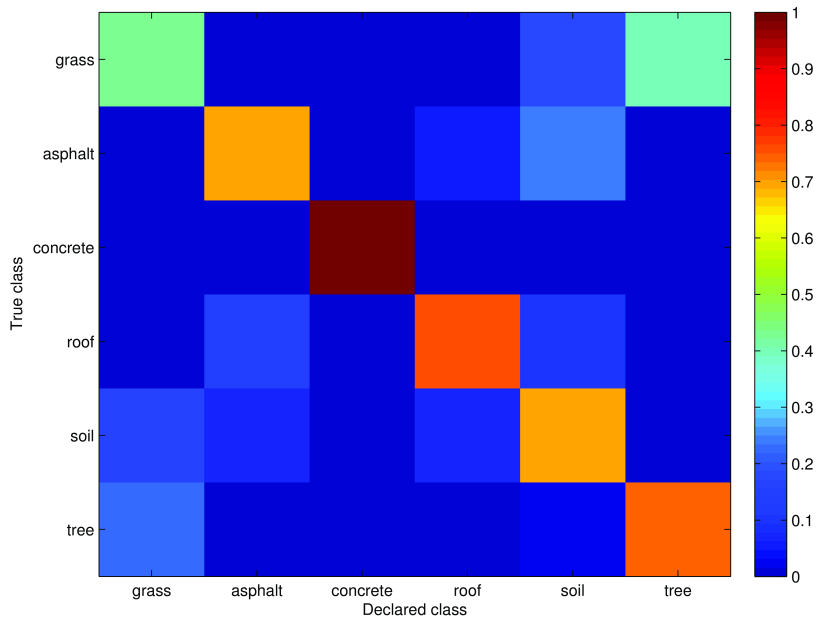


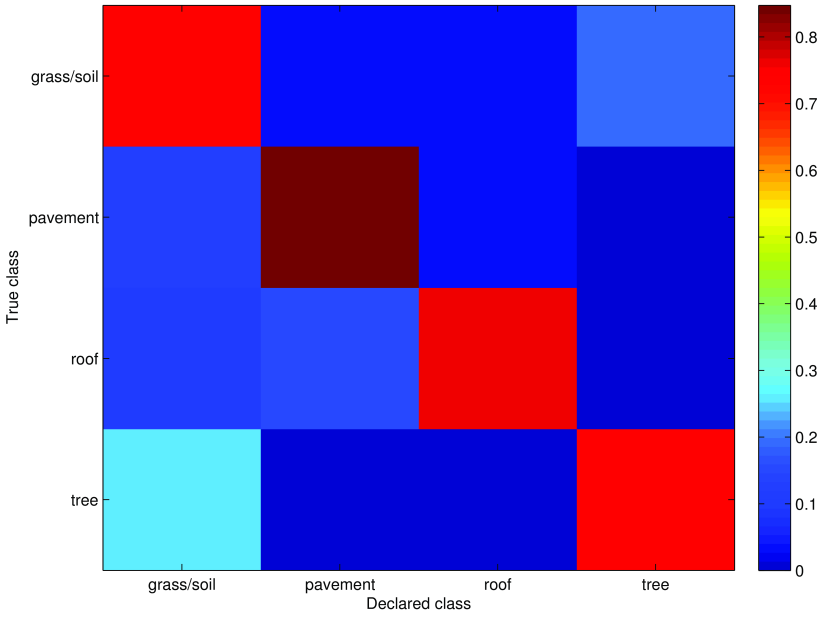
Figure 4.17: An illustration of the *a priori* spectral reflectance values used to train the adaptive background classifier. Each curve represents the mean of a population of samples for that particular class.

tion (3.29), an estimate of the transitional density function $p(\mathbf{z}|\phi)$ must be obtained through a confusion matrix analysis of the training data. This is achieved by presenting the fully trained AGRLVQI model with hetergenous mixtures of training samples and recording the resulting class declarations. After 100 trial mixtures of each true class, the confusion matrix illustrated in Figure 4.18 was obtained. This confusion matrix represents a reasonably successful ability to discern the functional classes. At this point the experiment initialization is complete and except for the simplifying fiducial, only *a priori* spectral signatures have been used. The experiment now transitions to mission-data and receives no further human input.

During the online, or mission stage, an adaptive HSI instrument such as the RITMOS would be addressed at each frame of time to select and measure a limited number of spectral pixels. The remaining pixels are measured as a panchromatic image. The adaptive background modeling uses exactly the same panchromatic image segmentation technique as the semi-automated background modeling. The resulting segmented regions were illustrated in Figure 4.13, and are used as the basis for this experiment. It is convenient to choose a single HSI cube, which has been rendered as if it were acquired in a single frame. Since the background and platform are static, this cube will serve for all subsequent frames, with no loss of experiment fidelity. Also for convenience, each pixel in the cube is immediately classified by the AGRLVQI model and stored in a cache of classifier output. After applying the functional material hierarchy, a labeled mapping is obtained, illustrated in Figure 4.19. This represents the potential end-state of the adaptive background modeling if it were allowed enough time to visit each pixel, and if the panchromatic segmentation had worked perfectly. A consequence of this segmentation on the attainable adaptive background model will be discussed momentarily. Given the sensor constraints, this full visitation would require 560 frames (one per line). Next, the segment class label probabilities are initialized to uniform distributions as in Equation (3.28), i.e., each segment has equal probability to be of any class. The iterative portion of the background modeling then begins. Each iteration k starts with updating the region entropy $h(k)$ (Equation (3.32)) and



(a) native classes



(b) functional classes

Figure 4.18: Results of the confusion matrix analysis for the adaptive background modeling technique. Each cell shows the ratio of times that a heterogenous mixture of a true class (row) was declared as a class (column) to the total number of times the true class was presented. Ideally, all declarations would lie on the diagonal. The native classes in (a) are reduced via the hierarchy to the final functional classes shown in (b).

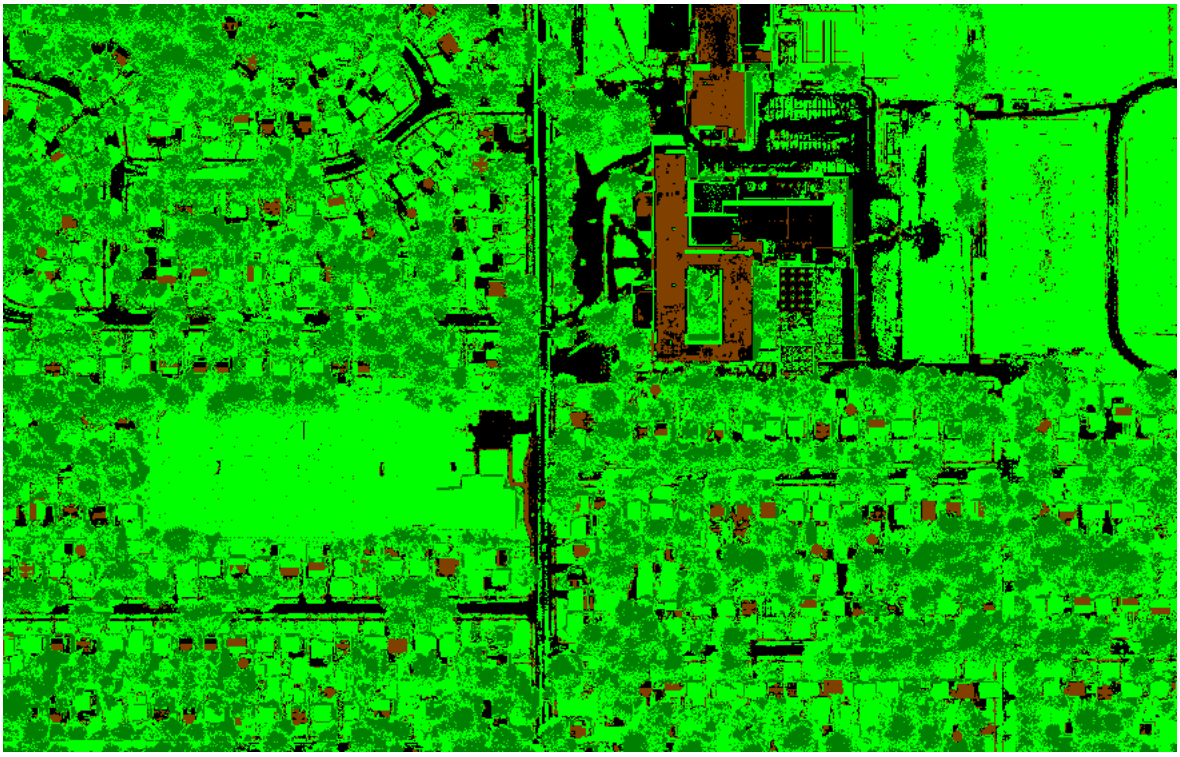


Figure 4.19: The map of class labels from the dense classification test. This represents the end state of the adaptive background modeling – given unlimited iterations and perfect region segmentation – where each pixel is measured and classified independently.

utility $u(k)$ (Equation (3.33), (3.34), or (3.35)). Based upon the maximum utility constrained by the sensor capability, e.g., requesting only one HSI pixel per column in the RITMOS instrument, the SRM determines which HSI pixels to collect. Each pixel is then incorporated into the region statistics according to Equation (3.30), and the *maximum a posteriori* class identity ϕ_{MAP} of each region is determined according to Equation (3.31).

The performance metric for the adaptive background modeling process is the probability of correct classification P_{CC}. For the sake of this evaluation, the labeled mapping in Figure 4.19 will serve as the truth. The metric P_{CC} is then the percentage of pixels of ϕ_{MAP} which match the corresponding truth pixel.

The three utility functions described in Section 3.4.3 have been tested in independent trials of the adaptive background modeling process. The resulting P_{CC} at each frame in time is shown in Figure 4.20. Recall that the adaptive background modeling process is intended to give the optimal *at-any-time* answer for $\phi_{\text{MAP}}(k)$ at any k. The results show a favorable initial P_{CC} followed by a rapid initial increase until a point of diminishing returns is reached. The u_{entropy} utility function shows some instability as the model matures, particularly after frame 50. This corresponds to regions of relatively large area transitioning between right and wrong $\phi_{\text{MAP}}(k)$. A consequence of the Bayesian ID process is that although the class probabilities for a region may change gently as new measurements are incorporated, the maximum *a posteriori* answer will change abruptly. The $u_{\substack{\text{norm} \\ \text{entropy}}}$ utility function has the best balanced performance with the fewest P_{CC} transients. Interestingly, the u_{random} performs quite well given it is random sampling, although it does suffer the most aggressive transients. This lends credence to the rule-of-thumb in compressive sensing that when Nyquist sampling of a space is too costly, a simple random sampling with intelligent reconstruction is a good minimal-effort substitute.

Next, a selection of frames from the normalized entropy SRM will be described. At the first frame, Figure 4.21, the system requests 880 HSI pixels (one per column)

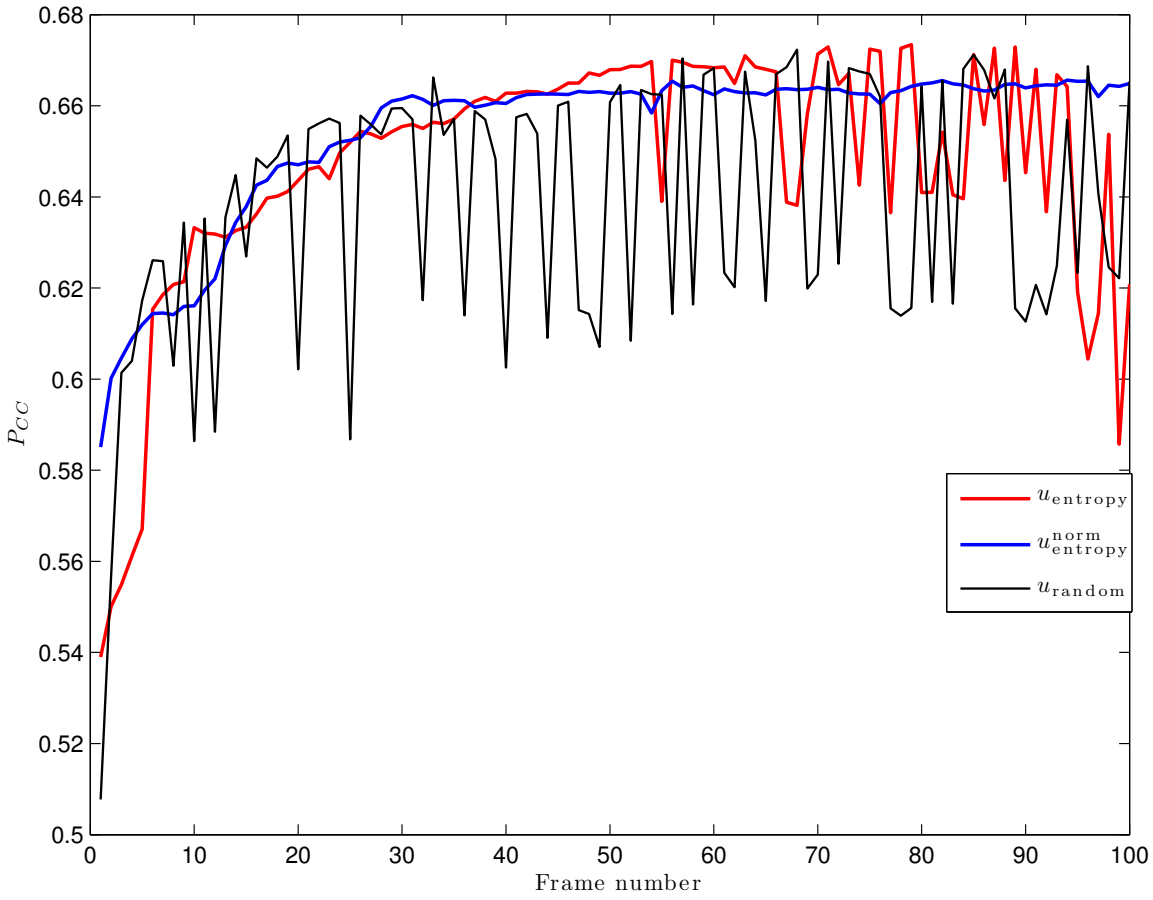


Figure 4.20: An illustration of the performance of the various adaptive background model SRMs. The three forms of the utility function u are the entropy, normalized entropy, and randomness.

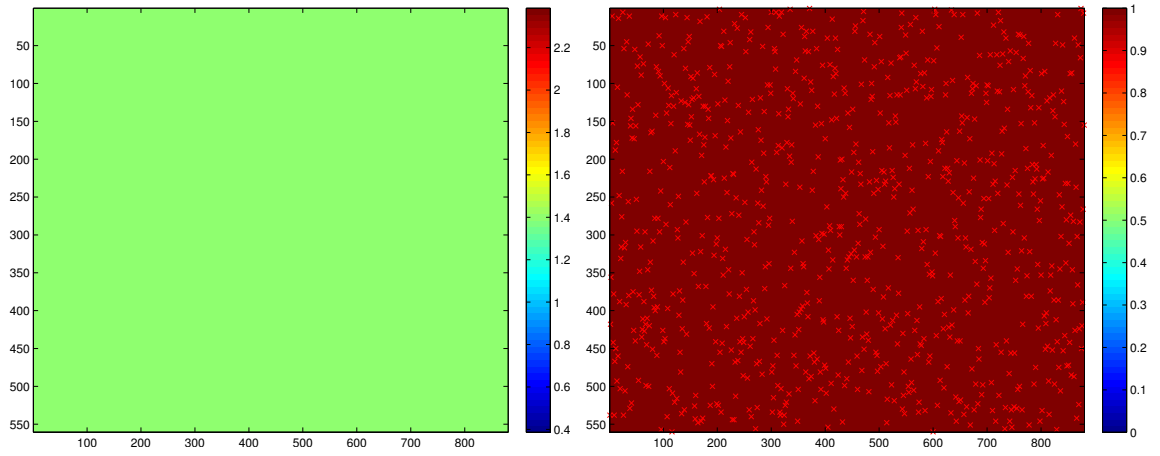
in a random fashion because the entropy and utility are initially uniform. The class labels are already sufficiently clear enough to discern portions of roads, buildings, and tree canopies. At this first frame – 0.1 seconds into the vignette – more than half of the almost one half million pixels have been correctly labeled.

At the second frame, Figure 4.22, the entropy and utility maps now contain a great deal of structure. The utility map in particular is almost bimodal: there are many regions which successfully classified in the first frame of measurements and hence have low utility. For example, the large soccer field in the middle of the frame, which appears dark blue in the utility map. The selected pixels are now uniformly distributed across the regions with high utility, and markedly absent on the regions which met with success in the first frame. The class labels have improved by two more percentage points, which is most noticeable in the more clearly defined tree canopies. Certain large regions with erroneous labels exist. The large roof (brown) region in the middle of the frame should be predominantly grass (green).

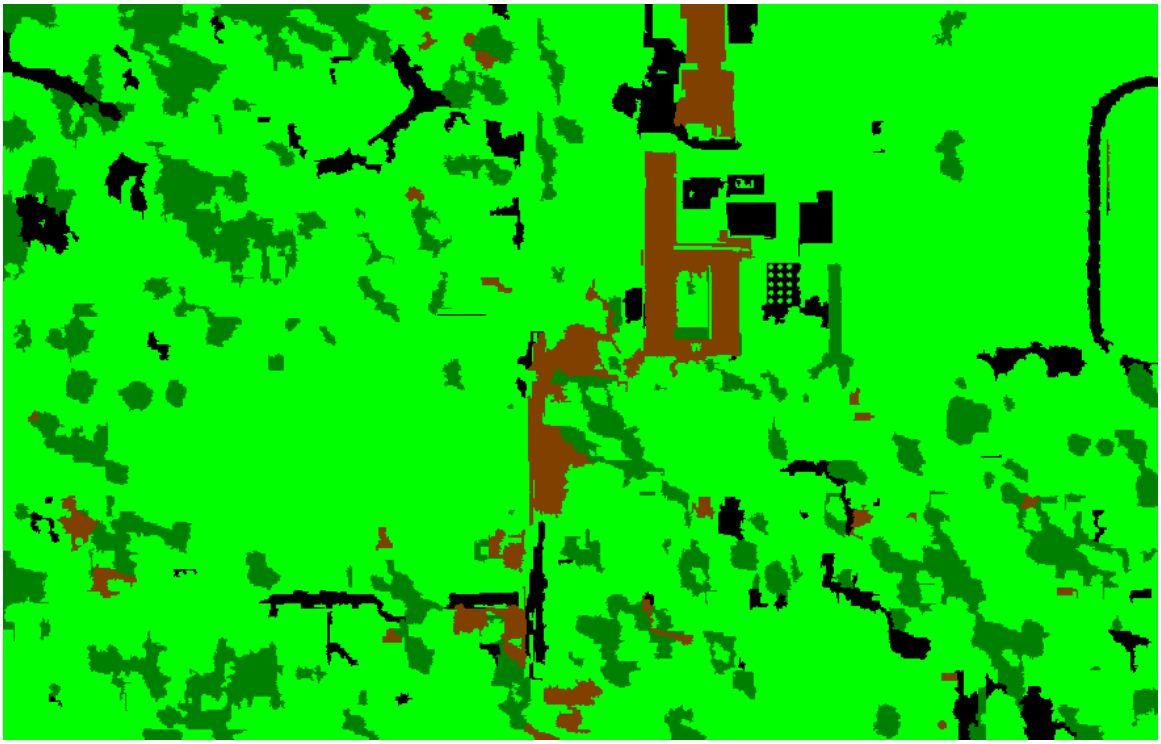
At the tenth frame, Figure 4.23, the entropy and utility maps have become more consistent. There are fewer large regions with extremely high utility. The SRM has naturally gravitated to refining smaller regions. The class labels have continued to improve, reaching $P_{CC} = 0.61$. Large erroneous regions still persist. It is insightful, however, to note the decrease in entropy from the second to tenth frame.

At the one-hundredth frame, Figure 4.24, the entropy and utility maps have become very uniform, with only troublesome regions still discernible. There are no large regions with high utility. This follows from the normalization term in Equation (3.34). The class labels have not improved significantly since frame 30, having reached $P_{CC} = 0.66$. There are few large erroneous regions in the class label map remaining. The large roof (brown) region in the middle of the frame was resolved in frame 23.

Recognizing that the original panchromatic spatial-segmentation approach has no hope of separating some materials, there will certainly be segments with het-



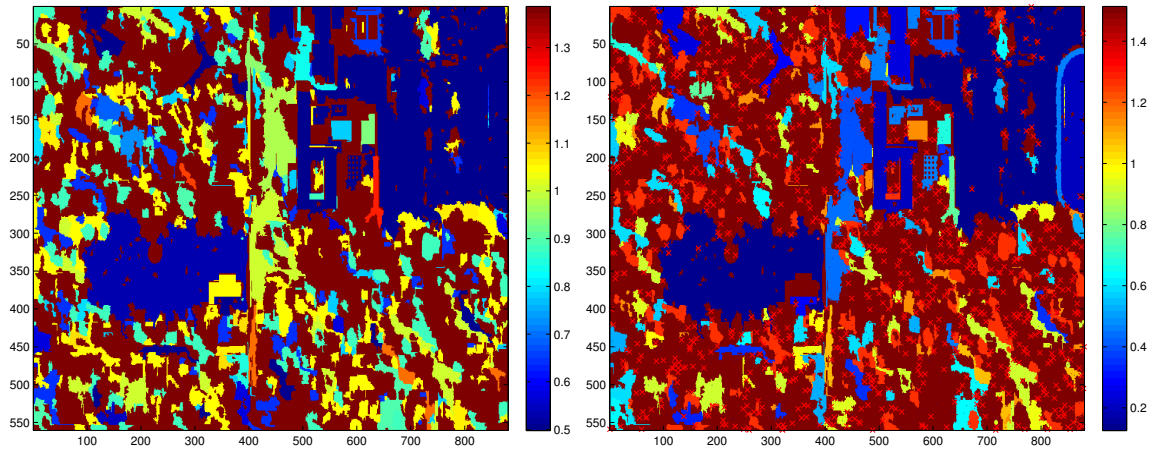
(a) entropy (b) utility



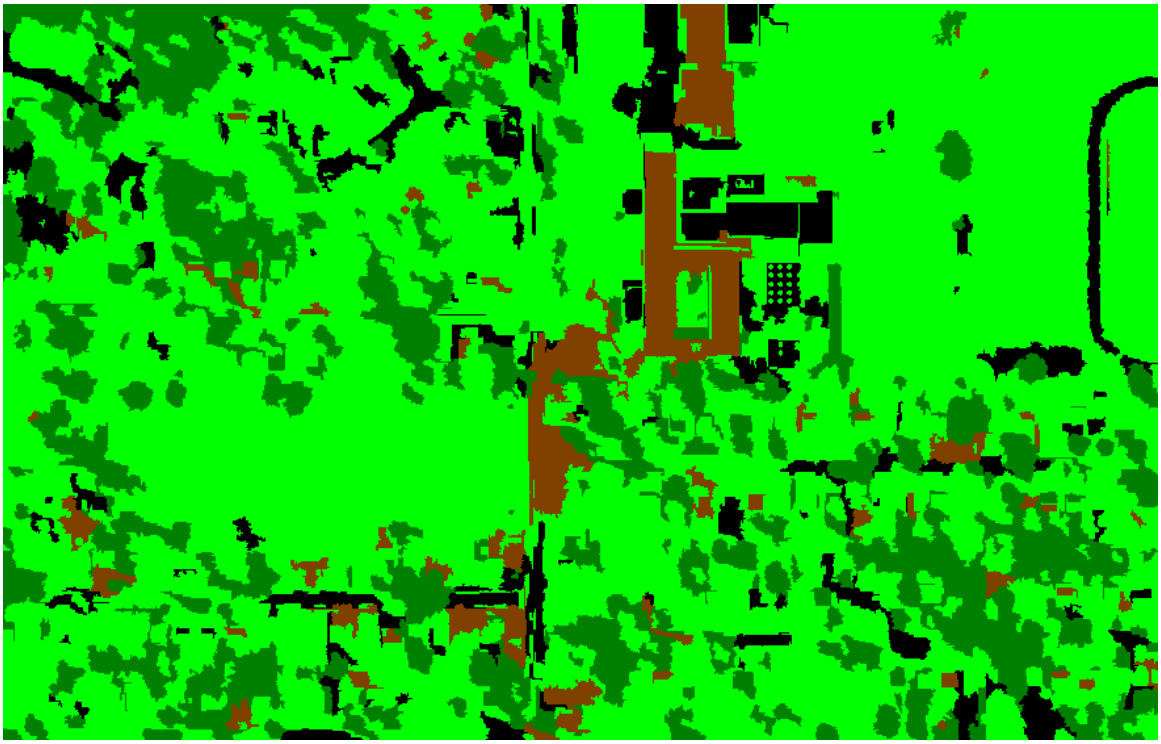
(c) class label

Grass Pavement Roof Tree

Figure 4.21: The adaptive background modeling results after one frame. The entropy (a) is initially uniform. The utility (b) is also uniform. The red crosses indicate the pixels which the SRM selected for HSI measurement at this frame. The *maximum a posteriori* class identity mapping shown in (c) already contains enough information to discern roads and buildings, with some notable errors.



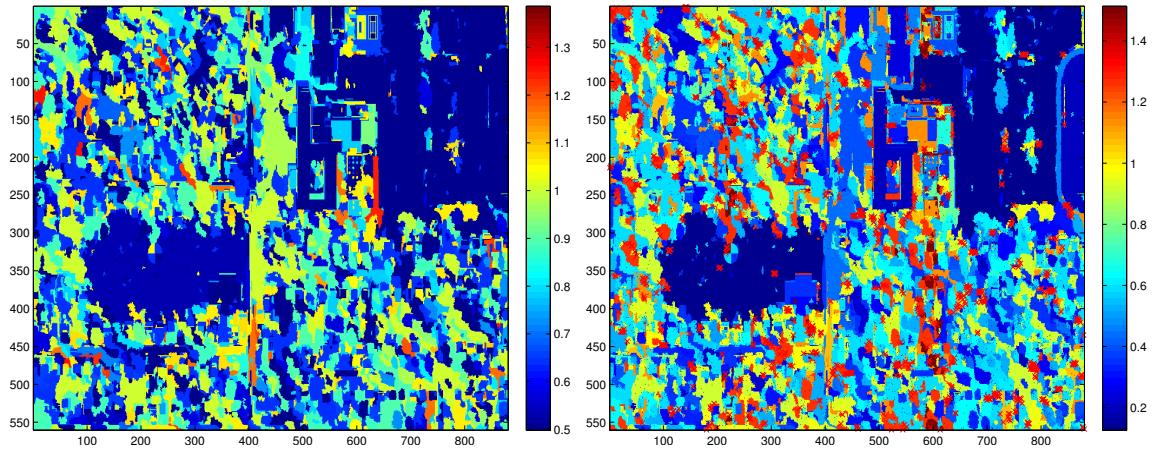
(a) entropy (b) utility



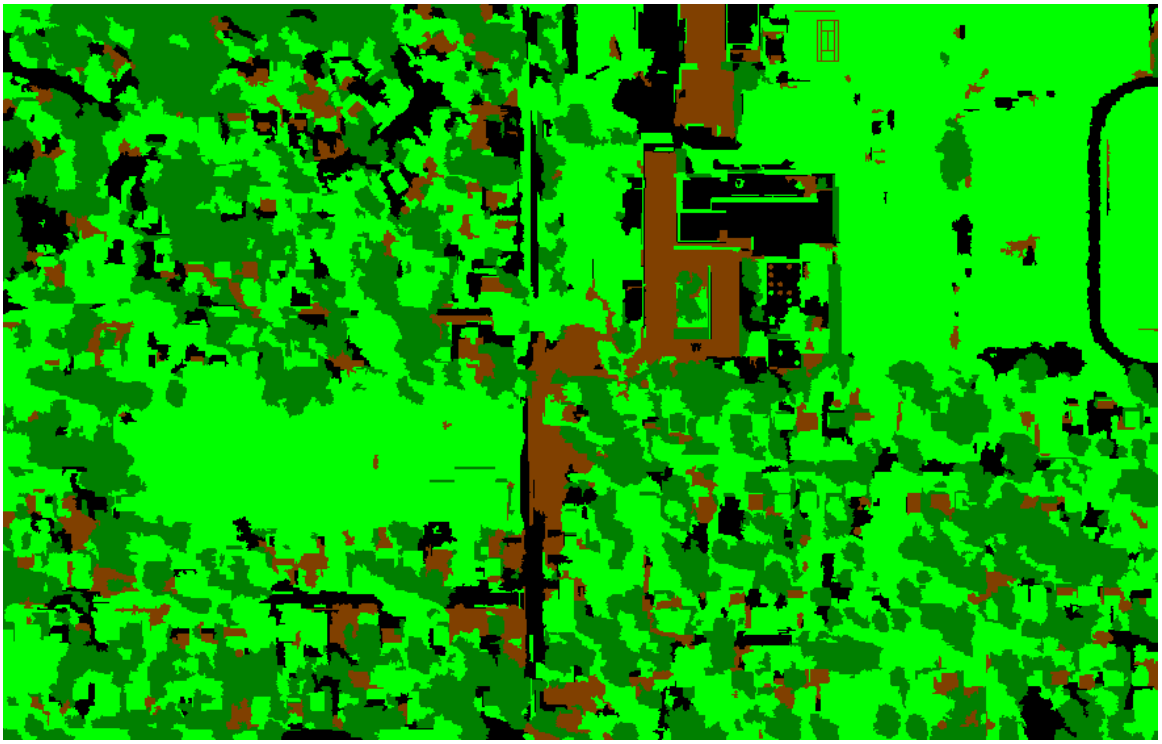
(c) class label

Grass Pavement Roof Tree

Figure 4.22: The adaptive background modeling results after two frames. The entropy (a) and utility (b) now show the structure of the scene. The *maximum a posteriori* class identity mapping shown in (c) is now demonstrating more refined tree canopy segmentation.



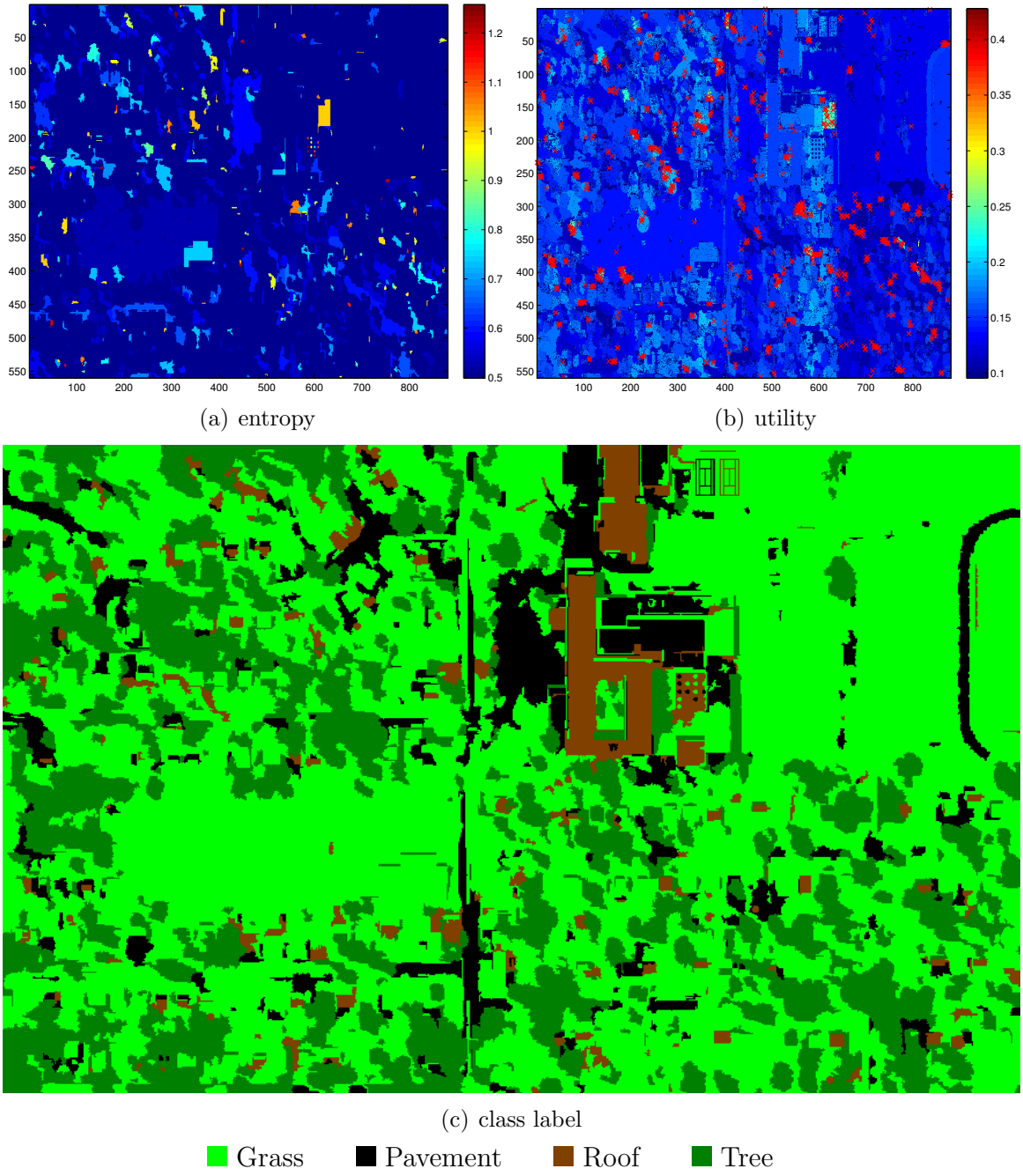
(a) entropy (b) utility



(c) class label

Grass Pavement Roof Tree

Figure 4.23: The adaptive background modeling results after ten frames. The entropy (a) and utility (b) are becoming more consistent, with fewer large regions of high value. The focus of the utility has shifted towards refining smaller regions. The *maximum a posteriori* class identity mapping shown in (c) has been further refined to resolve individual houses and portions of road between tree canopies.



(a) entropy

(b) utility

(c) class label

Figure 4.24: The adaptive background modeling results after 100 frames. The entropy (a) and utility (b) are now fairly uniform. The *maximum a posteriori* class identity mapping shown in (c) is now essentially as good as it will get. Dominant errors include single regions that clearly were mis-segmented by the spatial segmenter, and hence contain heterogenous HSI measurements.

erogenous materials – and hence, high entropy. As an example, consider a field of grass bisected by an asphalt road. Given the right conditions (grass health, asphalt weathering, and panchromatic sensor response), it is possible that the panchromatic spatial segmentation will not discern a difference, combining the field and road into a single segment. This problem could be mitigated by occasionally subdividing each segment having high entropy, despite a relatively large number of observations. The child regions ought to be chosen to have lower average entropy than their parent. Notably, child regions would inherit observations from their parent according to their new boundaries, such that history is not lost. Such a region splitting technique has not been implemented here, but would be a good candidate for a follow-on effort. The best attainable adaptive background model is clearly suboptimal – being capped at $P_{CC} = 66\%$ by frame 30 – due to this limitation. Or, a hybrid of the utility-function-based SRM and a fully-swept HSI cube could be utilized. In such a system, the adaptive sensor would be used according to the utility-function SRM for a certain small number of frames, e.g., 30, providing a useful background model in only three seconds of time. Then, the adaptive sensor could be commanded to measure all remaining HSI pixels in the scene, and a dense classification could be performed. In this way, and assuming stationary sensor field of view, the CAT system would have a $P_{CC} = 60\%$ answer in three seconds, and a $P_{CC} = 100\%$ answer within one minute. The entire process would remain autonomous.

4.2.4 Tracking Testbed. A tracking testbed has been augmented to serve as the CAT testing framework. This testbed is comprised of a mature MHT and associated logic, e.g., motion detector, association solver, coordinate conversions, Kalman filter and input/output management. It is primarily written in `Matlab`® , although certain portions have been ported to `C++` and used via the `Matlab`® extensions (MEX) interface. A screenshot taken from the testbed in use is shown in Figure 4.25. Several defining characteristics of the tracking testbed include:

- A pinhole camera model with an azimuth/elevation measurement space.

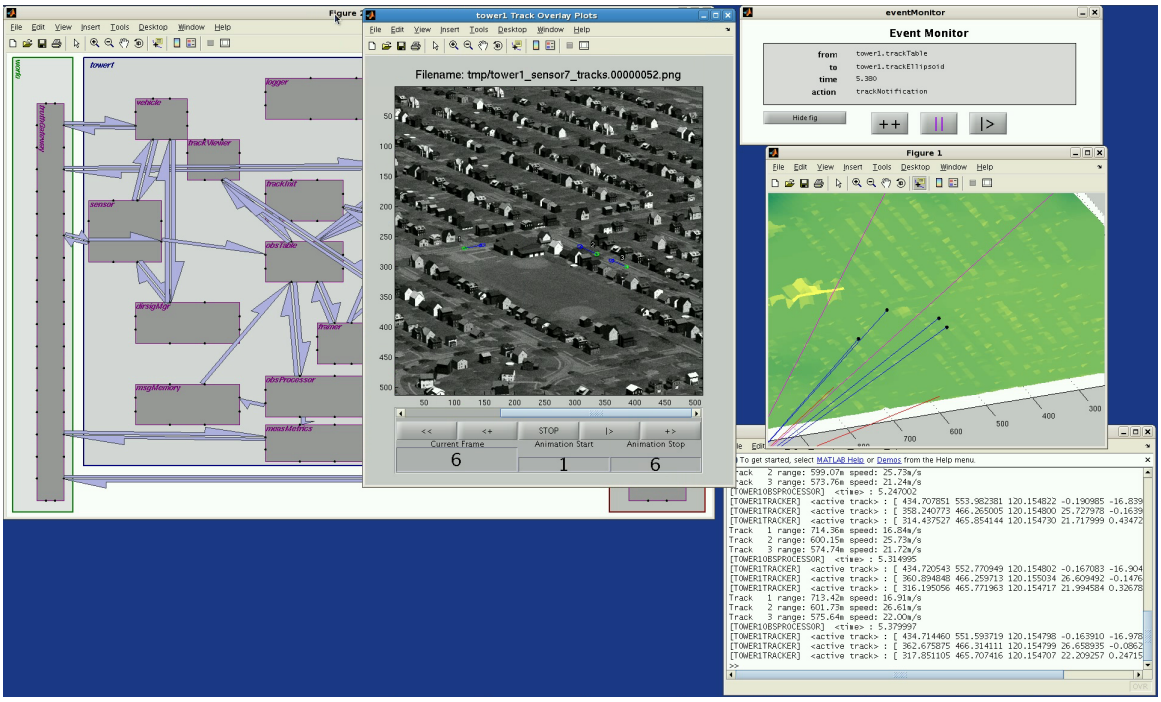


Figure 4.25: A screenshot of the tracking testbed in use.

- Digital terrain elevation data (DTED) derived terrain-intersection method for azimuth/elevation measurement augmentation into a three-dimensional measurement space.
- Internal easting, northing, up (ENU) state space with position, velocity, and acceleration components – a nine-dimensional state space.
- Nearly constant velocity target dynamics model.
- Single-frame track initiation, in which every measurement is given the chance to initialize a new track, even it if associated with a track or tracks in other hypotheses.
- A single MHT construct for both track initialization and track extension.
- The linear Kalman filter for state estimation.

4.2.5 Truth and Scoring. There are many methods for assessing the performance of a ground-vehicle tracking system. First, a precursory truth-to-track associ-

ation is performed. Truth is taken from the SUMO output. For each true vehicle i at time k, the sufficiently spatially close and valid tracks form the gated-set $\mathbf{G}(i,k)$. Valid tracks are those which have, at that time, already been confirmed and have not yet been dropped. A global-nearest-neighbor assignment is performed between the valid tracks and truth objects at each k. This assignment is mutually exclusive (done without replacement), and forms the function $\mathcal{I}(i,k)$ which either maps true object i at time k to a positive, natural, track identity ($\mathcal{I}(i,k) \in \mathbb{N}_1 \in \mathbf{G}(i,k)$) or to no track ($\mathcal{I}(i,k) = 0$.)

4.2.6 Metrics. The following is a set of well-known multi-target tracking metrics which have been identified as most likely to demonstrate the effects of CAT. The metrics in this section are extensions to the simpler parametric metrics defined in Section 4.1.1. Here, additional complexity is necessary to deal with the multiple-target, multiple-track nature of the tracking testbed.

Track completeness is defined as

$$\mathcal{M}_{\text{comp}}(k) = \frac{|i : \mathcal{I}(i,k) \neq 0|}{\mathrm{N}_{\text{true}}(k)} \ , \tag{4.8}$$

where $|\cdot|$ is the set counting operator used here to count the valid assignments, and $\mathrm{N}_{\text{true}}(k)$ is the number of true objects at time k. Notably, $\mathrm{N}_{\text{true}}(k)$ includes all tracks within the scenario area, including those which are occluded or stopped. Computing the average completeness from Equation (4.8) over all frames, $\bar{\mathcal{M}}_{\text{comp}}$, provides a measure of how well the tracker "covers" every true object with tracks throughout the scenario. It lies on the range $[0,1]$, where 1 indicates ideal coverage. As with the single target case, track identity is of no consequence to $\bar{\mathcal{M}}_{\text{comp}}$. Should a track drop and immediately be replaced by a new track on the same true vehicle, i.e., an identity-swap, $\bar{\mathcal{M}}_{\text{comp}}$ is not penalized.

Conversely, track purity is not concerned with coverage, but with track identity over the entire scenario $\mathbf{K}$:

$$\mathcal{M}_{\text{pure}}(i) = \frac{|k : \mathcal{I}(i,k) = \operatorname{mode} \mathcal{I}(i,\mathbf{K})|}{|k : \mathcal{I}(i,k) \neq 0|}, \tag{4.9}$$

which is the ratio of the frames in which a true object is assigned its most frequently occurring identity to the frames in which it is assigned any track identity. The aggregate purity is the same ratio extended to all true objects:

$$\bar{\mathcal{M}}_{\text{pure}} = \frac{\sum_{\forall i} |k : \mathcal{I}(i,k) = \operatorname{mode} \mathcal{I}(i,\mathbf{K})|}{\sum_{\forall i} |k : \mathcal{I}(i,k) \neq 0|}. \tag{4.10}$$

It lies on the range $(0, 1]$, where 1 indicates an identity assignment is entirely consistent. As $\bar{\mathcal{M}}_{\text{pure}} \to 0$, identity swapping occurs more frequently.

Track spuriousness is the ratio of tracks not spatially close to *any* true object, divided by the number of true objects. Recall that spatially close tracks are within the gated set $\mathbf{G}(i,k)$.

$$\mathcal{M}_{\text{spur}}(k) = \frac{\mathrm{N}_{\text{track}}(k) - |\bigcup_{\forall i} \mathbf{G}(i,k)|}{\mathrm{N}_{\text{true}}(k)}, \tag{4.11}$$

where $\mathrm{N}_{\text{track}}(k)$ is the number of tracks at time k. When $\mathcal{M}_{\text{spur}} = 0$, every track can be explained by a true object; when $\mathcal{M}_{\text{spur}} > 0$, some tracks are false-alarm tracks. The average spuriousness across all frames is $\bar{\mathcal{M}}_{\text{spur}}$.

Track redundancy is the ratio of tracks spatially close to *any* true object, divided by the number of true objects:

$$\mathcal{M}_{\text{redund}}(k) = \frac{|\bigcup_{\forall i} \mathbf{G}(i,k)|}{\mathrm{N}_{\text{true}}(k)}. \tag{4.12}$$

When $\mathcal{M}_{\text{redund}}(k) > 1$, some true objects are being overrepresented with extraneous tracks. The average redundancy across all frames is $\bar{\mathcal{M}}_{\text{redund}}$.

The cardinality of the truth, $\mathrm{N}_{\mathrm{true}}(\mathbf{K})$, is the number of unique truth objects in the entire scenario. The cardinality of the tracks, $\mathrm{N}_{\mathrm{track}}(\mathbf{K})$, is the number of unique identities assigned by the tracker over the entire scenario. The difference is

$$\Delta_{\mathrm{card}} = \mathrm{N}_{\mathrm{track}}(\mathbf{K}) - \mathrm{N}_{\mathrm{true}}(\mathbf{K}) \ , \tag{4.13}$$

which is the cardinality error, and is ideally 0.

4.2.7 Results. From the 100 seconds of synthetic data, three temporally non-overlapping vignettes of approximately 30 seconds each were identified. These vignettes were processed by the tracking testbed with a variety of parameter sets in uniform and CAT configuration. Splitting the data into vignettes was a matter of convenience, as it allowed for a threefold increase in the number of parallel testbed executions across a set of available computers. Each run of the tracking testbed on a single vignette required approximately 12 hours of computation time on a modern personal computer. This time accounts for all aspects of the testbed: detection, association scoring, MHT maintenance, image annotation, and metrics computation. For the uniform background statistics test, the P_{D} was identified as the first order statistic of influence. Therefore five values were selected for P_{D} (0.999, 0.970, 0.900, 0.800, and 0.500). The first three selections were made based on prior empirical experience in tuning this particular tracking system on other datasets. The remaining two selections were chosen as extended test cases. A set of testbed runs was performed for each. One additional set of testbed runs was performed with the testbed in CAT mode, operating according to the adaptive background statistics described in Section 3.4.4. The resulting metrics for both the context-aided and uniform tracking tests are given in Table 4.5.

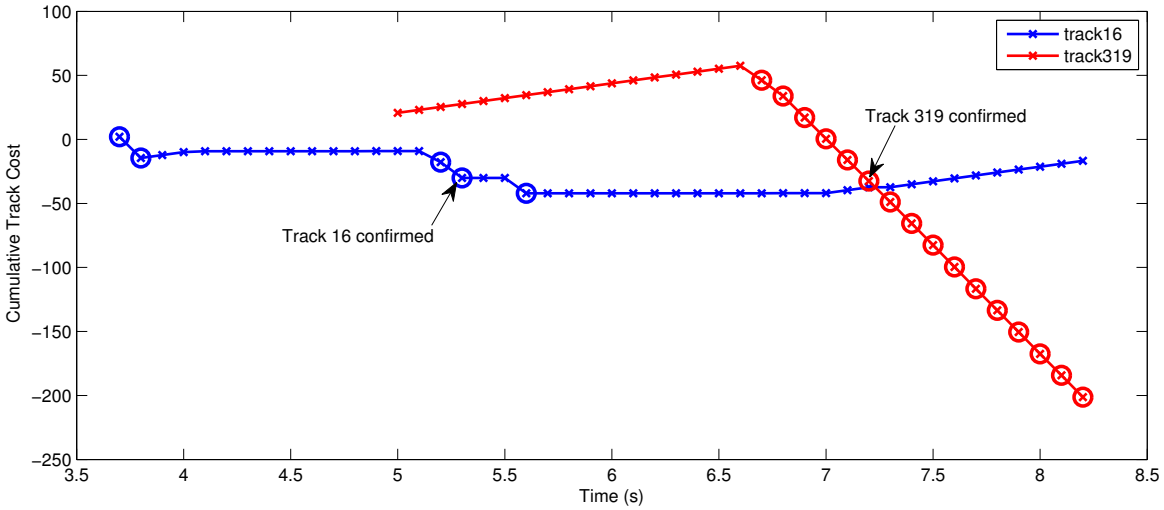
Several trends are apparent in the results. For the uniform background statistic runs, completeness improves as P_{D} decreases, suggesting that the tracker is confirming tracks quicker and deleting them slower. The penalty for such behavior comes in the form of deteriorating spuriousness and delta cardinality. This implies that as P_{D}

Table 4.5: Results for the tracking experiment. Three temporally non-overlapping vignettes were identified within the synthetic data. Five uniform parameter sets were created with the specified values for P_{D}. One additional parameter set was created with CAT statistics. The aggregate column represents the performance for the respective parameter sets over all three vignettes, and is calculated as the mean for $\bar{\mathcal{M}}$ metrics, and the sum for the remaining (counting) metrics.

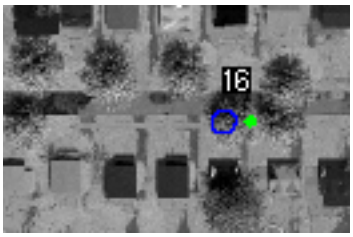
P_{D}	**0.999**				**0.970**				**0.900**			
vignette	**1**	**2**	**3**	**Aggr.**	**1**	**2**	**3**	**Aggr.**	**1**	**2**	**3**	**Aggr.**
$\bar{\mathcal{M}}_{\mathrm{comp}}$	0.6458	0.5824	0.7377	**0.6553**	0.7426	0.6651	0.7998	**0.7358**	0.7903	0.7000	0.8466	**0.7790**
$\bar{\mathcal{M}}_{\mathrm{pure}}$	0.7957	0.8220	0.8237	**0.8131**	0.7912	0.7939	0.8091	**0.7981**	0.7967	0.7722	0.8151	**0.7947**
$\bar{\mathcal{M}}_{\mathrm{spur}}$	0.0130	0.0223	0.0204	**0.0186**	0.0184	0.0269	0.0214	**0.0222**	0.0186	0.0269	0.0238	**0.0231**
$\bar{\mathcal{M}}_{\mathrm{redund}}$	0.6473	0.5841	0.7411	**0.6575**	0.7444	0.6683	0.8061	**0.7396**	0.7926	0.7062	0.8537	**0.7842**
$\mathrm{N}_{\mathrm{track}}(\mathbf{K})$	61	69	90	**220**	67	78	97	**242**	66	84	99	**249**
$\mathrm{N}_{\mathrm{true}}(\mathbf{K})$	47	54	60	**161**	47	54	60	**161**	47	54	60	**161**
Δ_{card}	14	15	30	**59**	20	24	37	**81**	19	30	39	**88**
P_{D}	**0.800**				**0.500**				**CAT**			
vignette	**1**	**2**	**3**	**Aggr.**	**1**	**2**	**3**	**Aggr.**	**1**	**2**	**3**	**Aggr.**
$\bar{\mathcal{M}}_{\mathrm{comp}}$	0.8132	0.7128	0.8613	**0.7958**	0.8495	0.7512	0.8819	**0.8275**	0.8050	0.7143	0.8582	**0.7925**
$\bar{\mathcal{M}}_{\mathrm{pure}}$	0.8020	0.7753	0.8278	**0.8017**	0.7821	0.7759	0.8116	**0.7899**	0.7498	0.7823	0.6827	**0.7383**
$\bar{\mathcal{M}}_{\mathrm{spur}}$	0.0227	0.0288	0.0262	**0.0259**	0.0304	0.0297	0.0322	**0.0308**	0.0441	0.0336	0.0452	**0.0410**
$\bar{\mathcal{M}}_{\mathrm{redund}}$	0.8167	0.7216	0.8714	**0.8032**	0.8544	0.7607	0.8936	**0.8362**	0.8243	0.7375	0.8978	**0.8200**
$\mathrm{N}_{\mathrm{track}}(\mathbf{K})$	69	83	101	**253**	73	87	109	**269**	78	86	125	**289**
$\mathrm{N}_{\mathrm{true}}(\mathbf{K})$	47	54	60	**161**	47	54	60	**161**	47	54	60	**161**
Δ_{card}	22	29	41	**92**	26	33	49	**108**	31	32	65	**128**

decreases, an increasing number of the tracks are invalid. Since redundancy is tightly following completeness in all runs, the tracker is apparently not creating redundant tracks in close proximity to true targets. Finally, the purity metric is relatively stable across the uniform background statistic runs. This suggests that in a uniform background statistic system, an improvement in purity is unlikely to be achieved by tuning P_{D}. In contrast to empirical experience on other datasets, the tested's performance among the uniform background statistic runs had the best balanced performance at $P_{\mathrm{D}} = 0.800$.

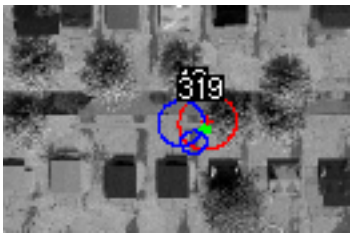
Upon inspection of the CAT results, the high cardinality error coupled with lower than expected purity and completeness is immediately apparent. A frame-by-frame inspection of the annotated CAT imagery indicated an unanticipated effect of CAT-based track scoring with respect to measurement-to-track association. A relatively frequent occurrence in the results is a post-occlusion track swap effect, an example of which is given in Figure 4.26. In many cases, the CAT system successfully coasted tracks through low P_{D} occlusions, only to swap them with a new track immediately after measurements resumed post-occlusion. Recalling that this tracking system uses a brute-force, single-frame track initialization technique, there are always numerous new tracks present. These immature tracks are hypothetical initializations, but are generally not confirmed, and therefore do not confuse the operator or reduce tracking metrics. However, should one of these immature tracks exist in a high P_{D} region nearby a coasting mature track which is in a low P_{D} region, the tracks will be in competition for new measurements post-occlusion. The mature coasting track may register as being in a low P_{D} condition for several frames after the occlusion is over due to error in the background model or error in the state estimate of the target. Due to the CAT statistics – by design – it is extremely inexpensive in terms of association cost to continue coasting such a track without measurements. Conversely, the immature false track in a high P_{D} region is somewhat more expensive in terms of association cost to coast without measurements, regardless of its poor health. The convergence of these causalities is not guaranteed after each occlusion. However,



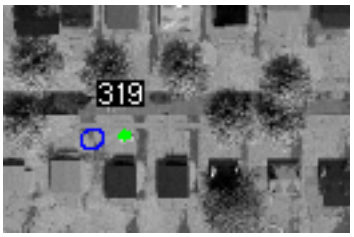
(a) Cost versus time for the tracks in question. Circles indicate measurement-to-track associations. Track 16 initializes at time 3.60s from "lucky" detections through a dense tree canopy – therefore a low P_{D} region. Track 319 initialized from a nearby false alarm outside of the canopy – a high P_{D} region. Note that prior to time 5.00s, track 319 is not shown in this plot as it existed in a hypothesis other than the top hypothesis. Track 319 coasts without measurements until time 6.70s when it begins to "steal" measurements from track 16. This represents the "unfair" advantage that unhealthy, unconfirmed, coasting tracks in high P_{D} regions hold over any nearby track in a low P_{D} region.



(b) Time = 5.60s. Track 319 has not yet been confirmed, so does not appear in the track output report. This is the last measurement that track 16 will obtain. The green dot is the current position estimate for track 16. The blue ellipse is the one-second prediction with covariance for track 16.



(c) Time = 7.20s. Track 319 has just confirmed, and has "stolen" many measurements from track 16 already due to its higher P_{D}. Here, the red ellipse is the current position estimate with covariance for the unhealthy track 16. The larger blue ellipse is the one-second prediction with covariance for track 16. The green dot is the current position estimate for track 319. The smaller blue ellipse is one-second prediction with covariance for track 319.



(d) Time = 9.60s. Track 16 has just been deleted due to a high track cost. Track 319 has updated with all post-occlusion measurements.

Figure 4.26: An illustration of the CAT cost offset problem.

should they happen, it seems that the proper track will always lose the post-occlusion fight for measurements, allowing the immature track to confirm and "steal" all future measurements while the proper track is deleted. This results in a track identity swap, and can have a severe and negative impact on the track purity measurement. Since the MHT is attempting to optimize the combined association costs, this undersirable result is not a coincidence and can be directly attributed to the method used to compute the association costs. As such, a mitigation method was developed as described next.

While this track cost problem is not specifically systemic to the CAT system, it does seem to be exacerbated by the presence of tracks with differing P_{D} values – a situation which never arises in the uniform-statistics system. This problem forced a recall of an early design decision in which a dual-stage MHT system was considered, wherein all measurements are first used to possibly extend confirmed tracks within the extension MHT, and leftover measurements are then sent to the initialization MHT to create new tracks. In such a system, the confirmation process involves porting a track from the initialization MHT into the extension MHT. A primary argument for this approach is to reduce computational complexity by using suboptimal tuning parameters in the initialization MHT, especially by reducing hypothesis tree depth. Recall that the depth of the hypothesis tree increases the deferred decision-making capability of the MHT and has the potential to improve firm decisions. However, this comes at significant computational cost since the number of hypotheses is growing expoentially. A secondary argument for the dual-MHT approach is to segment the confirmed tracks from the initializing tracks and give the confirmed tracks an advantage while competing for measurements. This dual-MHT design was initially rejected as overly complex with respect to algorithmic design, and the single-MHT implementation was used instead with good results. This track cost problem, though, motivates the need to favor confirmed tracks in the measurement-to-track association solution. Therefore, an *ad hoc* confirmed track cost offset has been proposed and implemented to mimic the desirable effects of the dual-MHT solution without the ad-

ditional algorithmic complexity. Simply, the negative-log-likelihood cost modification $\Delta C(k)$ given in Equation (3.12) is modified to become

$$\Delta C(k) = \begin{cases} -\ln\left[1 - P_{\mathrm{D}}\right] & \text{miss} \\ -\ln\left[\frac{P_{\mathrm{D}} p(z_k^j | x_k^i)}{\beta_{\mathrm{FA}}}\right] & \text{update non-confirmed track} \\ -\ln\left[\frac{P_{\mathrm{D}} p(z_k^j | x_k^i)}{\beta_{\mathrm{FA}}}\right] - 5 & \text{update confirmed track} \end{cases}, \tag{4.14}$$

where the firm-track-offset of 5 was empirically selected, but is related to the slope of the costs seen in Figure 4.26.

Repeating the CAT experiment with Equation (4.14) in force resulted in a subjective elimination of the post-occlusion track-identity swap problem. It is reasonable to assume that – given the presence of heterogenous P_{D} values – this change would affect CAT more significantly that the uniform-statistics system. In order to test the validity of this claim, the uniform test with the best balanced performance was chosen ($P_{\mathrm{D}} = 0.800$) and repeated with Equation (4.14) in force. The resulting metrics from both repeated experiments are given in Table 4.6. Here, the CAT system has benefited from the cost offset technique; the uniform background statistics system has improved slightly. The most significant change is the delta cardinality for the CAT system, which as improved from a 128 track deviation to only a 44 track deviation. Comparing the delta cardinality of the CAT system (44) to that of the best observed uniform background statistics system (63) indicates that the CAT system was closer to the true cardinality of the scenario by 19 tracks. This is a 30% reduction in extraneous track identities, and represents 19 times in which the CAT system properly maintained a target's identity. The completeness and purity metrics for the CAT system have improved to a point 4% beyond that of the best observed uniform background statistics system.

Recalling the findings in Section 4.1.3, there are portions of the OC space in which CAT is predetermined to be of no help, such as un-occluded segments of road. The vignettes utilized here include a mixture of un-occluded roads, short occlusions

Table 4.6: Results for the tracking experiment with a firm-track cost-offset applied. Three temporally non-overlapping vignettes were identified within the synthetic data. A single uniform parameter set with $P_{\mathrm{D}} = 0.800$ was selected to represent the best uniform case. One additional parameter set was created with CAT statistics. The aggregate column represents the performance for the respective parameter sets over all three vignettes, and is calculated as the mean for $\bar{\mathcal{M}}$ metrics, and the sum for the remaining (counting) metrics.

P_{D}	**0.800**				**CAT**			
vignette	**1**	**2**	**3**	**Aggr.**	**1**	**2**	**3**	**Aggr.**
$\bar{\mathcal{M}}_{\mathrm{comp}}$	0.8262	0.7302	0.8697	**0.8087**	0.8667	0.7593	0.9065	**0.8442**
$\bar{\mathcal{M}}_{\mathrm{pure}}$	0.8344	0.8624	0.8388	**0.8452**	0.8838	0.9019	0.8404	**0.8754**
$\bar{\mathcal{M}}_{\mathrm{spur}}$	0.0281	0.0307	0.0318	**0.0302**	0.0498	0.0380	0.0463	**0.0447**
$\bar{\mathcal{M}}_{\mathrm{redund}}$	0.8288	0.7345	0.8763	**0.8132**	0.8730	0.7645	0.9253	**0.8543**
$\mathrm{N}_{\mathrm{track}}(\mathbf{K})$	62	69	93	**224**	59	61	85	**205**
$\mathrm{N}_{\mathrm{true}}(\mathbf{K})$	47	54	60	**161**	47	54	60	**161**
Δ_{card}	15	15	33	**63**	12	7	25	**44**

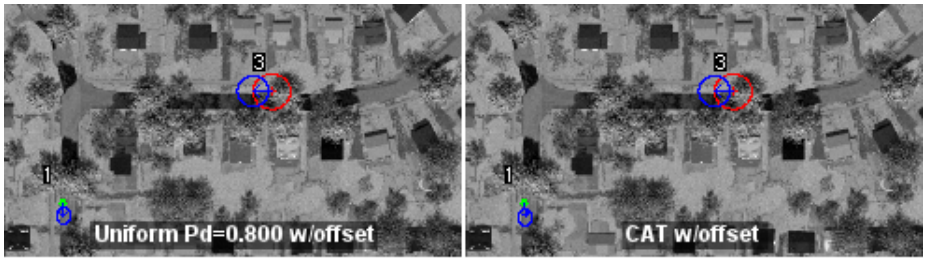
where CAT is unnecessary such as a single tree canopy, and heavily occluded segments where the CAT system was the only observed technique for maintaing track identity. Since the completeness and purity metrics are calculated over the entirety of the vignettes – including the un-occluded portions – the 4% performance increase is not insignificant. In contrast, the delta cardinality metric is a *de facto* judge of performance in the most difficult portions of the vignettes. A track deletion in a heavily occluded region results in one additional penalty to the delta cardinality regardless of how challenging or benign the remainder of the vignette may be. Therefore, the delta cardinality improvement of the CAT system over uniform background statistics by 30% is a justifiable first-order result.

In order to illustrate the impact of CAT, selected excerpts from the annotated output images are presented in a storyboard fashion. In Figure 4.27, a target traverses a road segment with two medium-length occlusions over a span of six seconds. The CAT system successfully maintains the track through both occlusions, while the uniform background statistics system deletes the track both times.

In Figure 4.28, a number of targets travel in opposing directions on a two lane road. This excerpt is especially challenging due to an occlusion that covers only one lane of the road. This creates additional ambiguity for the measurement-to-track association solver. The CAT system is inferior to the uniform system in this excerpt due to a delayed track confirmation with cascading effects. This is an example of unintended consequences; normally a delay in track confirmation does little harm, but here it causes track identity loss.

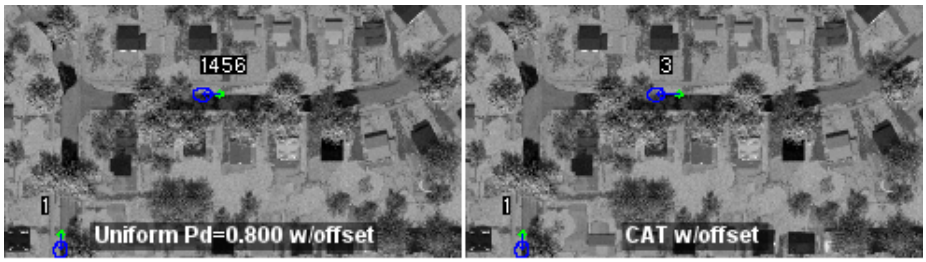
In Figure 4.29, two targets approach each other on a two lane road. Each target encounters a medium-length occlusion. The uniform background statistics version deletes one of the tracks during its occlusion, which permits a subsequent identity swap with the other vehicle. The CAT system, however, is able to coast both tracks through the occlusions and maintain track identity.

Finally, in Figure 4.30 the longest observed occlusion from all vignettes incorporates a road covered in dense tree canopy. The CAT system is able to maintain the track through the $10s$ event, whereas the uniform background statistic system deletes the track $1.9s$ into the occlusion.



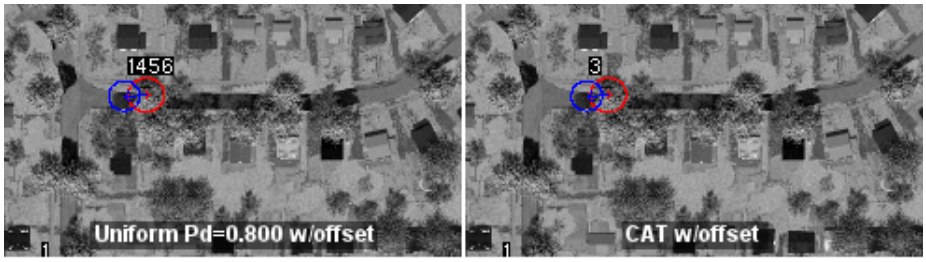
vignette: 2, 43.00 s

(a) The same frame of imagery from both the uniform (left) and CAT (right) tests. Firm-track cost-offset is applied in both cases. The current position of each track is shown in green if it has associated with a measurement on this frame and red if it has not. The one-second future prediction of track location is shown in blue. Elipse sizes indicate uncertainty. Here, track 3 has entered occlusion due to tree canopy. On the next frame, the uniform system deletes track 3.



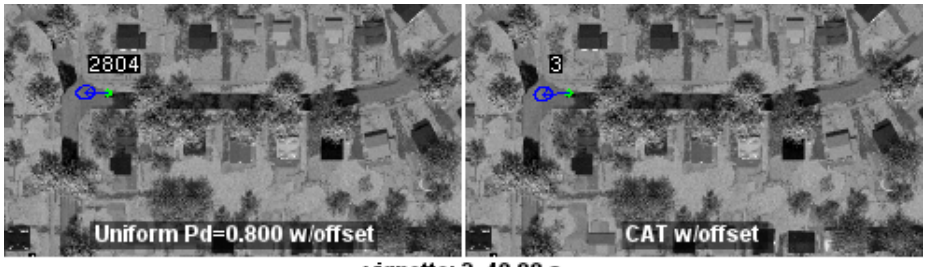
vignette: 2, 44.80 s

(b) The CAT system has coasted track 3 through the occlusion. The uniform system has just initialized a new track 1456 on the same object.



vignette: 2, 47.90 s

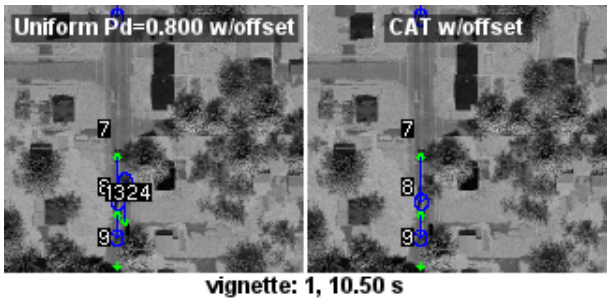
(c) Again, the CAT system is coasting track 3 through another occlusion. The uniform system will delete track 1456 on the next frame.



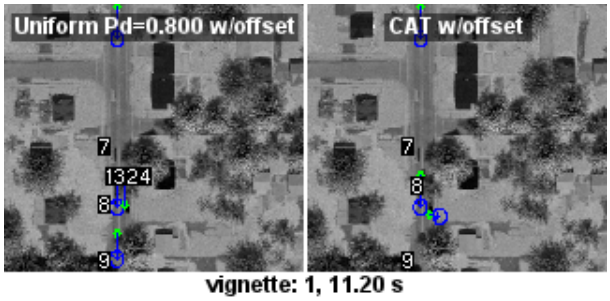
vignette: 2, 49.00 s

(d) This example ends with the CAT system having successfully tracked the object through two occlusions without loss of identity. The uniform system has swapped IDs twice.

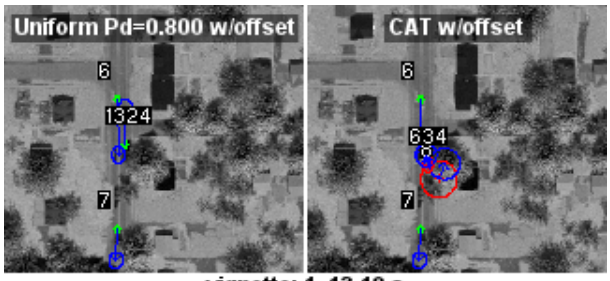
Figure 4.27: An illustration of CAT maintaining track identity, whereas the uniform system incurs multiple instances of track identity loss.



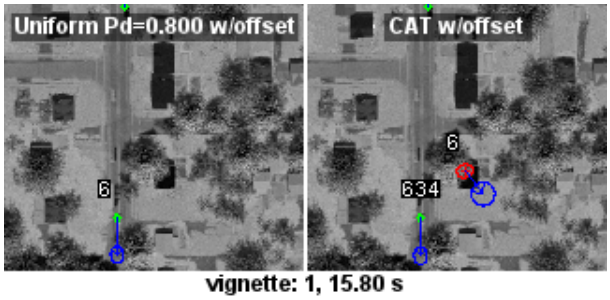
(a) The same frame of imagery from both the uniform (left) and CAT (right) tests. Firm-track cost-offset is applied in both cases. Here, tracks 7, 8, and 9 are Southbound. A Northbound vehicle has just emerged from a tree and has been confirmed by the uniform system as track 1324.



(b) The lack of confirmed track for the Northbound vehicle has proven disastrous for the CAT system. When the Southbound track 8 missed 3 subsequent detections near time 10.8s, the track began to steer onto the Northbound vehicle. Here, track 8 has steered significantly Eastbound.



(c) The CAT system has finally confirmed track 634 for the Northbound vehicle, but it competes with track 8 for measurements. Note the Southbound track 6.



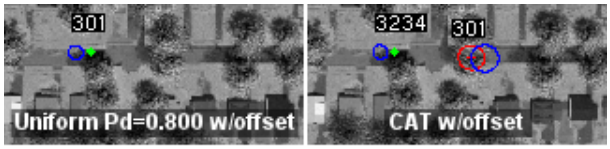
(d) Finally, the CAT track 8 is deleted due to poor scores, and tracks 6 and 634 swap in the midst of poor scores and high state covariance.

Figure 4.28: An illustration of CAT performing poorly compared to the uniform system. Here, a delay in track confirmation (CAT) results in a cascaded set of track swaps.

(a) The same frame of imagery from both the uniform (left) and CAT (right) tests. Firm-track cost-offset is applied in both cases. Here, an Eastbound vehicle exists at track 301 in both systems. A Westbound vehicle is track 1891 in the uniform system, and 3234 in the CAT system. Both vehicles are just about to enter occlusions.

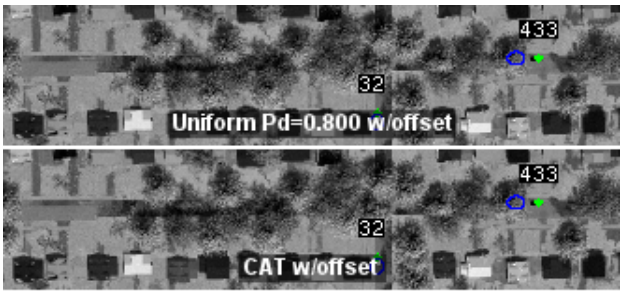
(b) Both vehicles emerge from the respective occlusions. However, the uniform system has deleted the track for the Westbound vehicle.

(c) Here, the Eastbound vehicle does not produce a measurement (note red circle in the CAT system). However, the uniform system has allowed track 301 to "steal" the measurement from the Westbound vehicle. This begins a track identity swap.



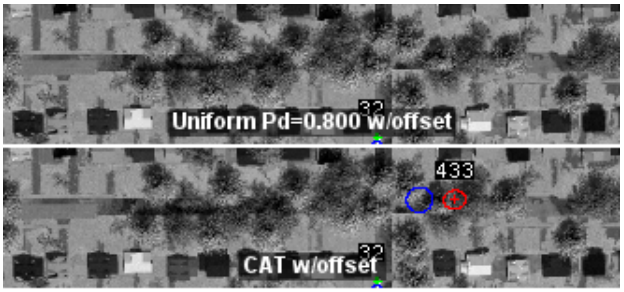
(d) Finally, the uniform system has lost the Westbound vehicle and continued its identity on the Eastbound vehicle erroneously. The CAT system has maintained track identities.

Figure 4.29: An illustration of CAT preventing an identity swap by successfully coasting a track through occlusion.



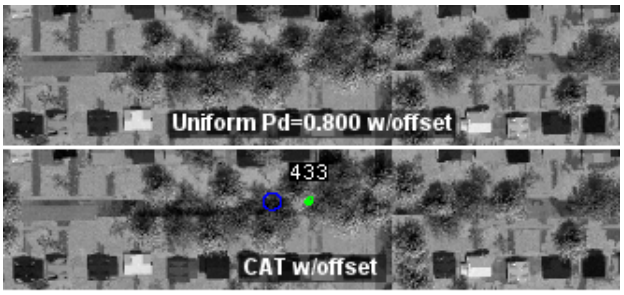
vignette: 3, 67.70 s

(a) The same frame of imagery from both the uniform (top) and CAT (bottom) tests. Firm-track cost-offset is applied in both cases. Here, a Westbound vehicle exists as track 433 in both systems.



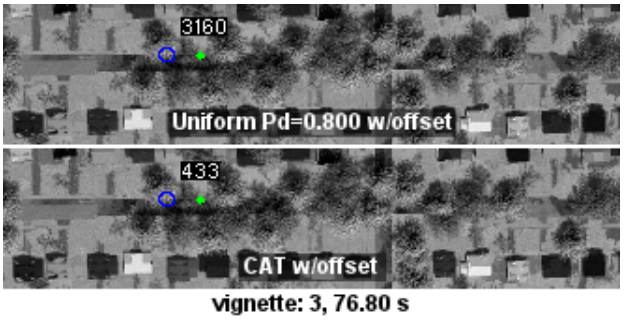
vignette: 3, 70.10 s

(b) The vehicle has entered a long occlusion – 10s – in which it will be detected 26 times and missed 74 times. The uniform system has deleted the track 1.9s into the occlusion.



vignette: 3, 73.70 s

(c) An example of a frame in which the vehicle generated a measurement. This reduces the track cost in the CAT system, reinvigorating the track. It is of little use in the uniform system, since it is insufficient to form a new track.



vignette: 3, 76.80 s

(d) Near the end of the occlusion, several concurrent detections permit the uniform system to finally form a new track on the vehicle. However, track identity loss has already occurred. The CAT system successfully maintains the track identity.

Figure 4.30: An illustration of CAT preventing track deletion through an extended occlusion.

V. Conclusions

In this chapter, the contributions and results of the effort will be reviewed in order to draw several important conclusions. First and foremost,

Does the novel context-aided tracking system outperform the current state-of-the-art tracking system with uniform background statistics? Decidedly, yes.

The evidence to support this statement begins with the fundamental tracking theory in Section 3.2. Accepted theory and practice supports the concept of track costs based upon the statistics of the measurement system, which by virtue of occluding elements are dominated by the background statistics themselves.

A compelling and comprehensive analysis was performed with a parametric tracking simulator described in Section 4.1. The key finding from that experiment is that – once isolated into a simplifying single-target situation – CAT improves track purity and tracker cardinality by as much as 50% and completeness by as much as 400%. Also insightful is that such performance increases are conditioned upon extremely difficult scenarios, e.g., frequent and long occlusions; in benign conditions, CAT neither helps nor harms the system.

Next in the chain of evidence, Section 3.4 presented two approaches with which to develop a background model of the scene, identifying key functional elements with the use of hyperspectral data. A semi-automated approach qualitatively arrived at a very reasonable background model with minimal operator intervention. A novel, adaptive, purely autonomous approach – a key contribution of this effort – was presented in Section 3.4.2 and supported with a new SRM technique in Section 3.4.3. In Section 4.2.3, this SRM was tested and proven to converge to a 66% correct adaptive background model in $\frac{1}{18}^{\text{th}}$ the time of a non-adaptive approach – a 95% reduction in sensor acquisition time. A hybrid technique was suggested which transitions from the 66% answer to the 100% answer as soon as it has been fully acquired, resulting in a zero-latency model with a full-performance steady-state.

The final evidence for the efficacy of CAT was given in the form of a high-fidelity tracking testbed. A mature and relevant MHT was augmented to perform CAT with the semi-automated background model in Section 4.2. An important finding regarding the impact of CAT on single-stage MHT systems was uncovered; a compact and effective solution was given in the form of a firm-track-cost-offset. This is another key contribution to the body of tracking research. The final analysis showed that CAT improved the completeness and purity of the tracking testbed by 4% over uniform background statistics. While modest, this finding is justifiably defended by the range of complexity of the tracking vignettes, i.e., the performance gain is diluted by benign target activity much of the time. Another metric, the delta cardinality, has been proposed as the most salient measure of performance gain in the CAT system. It directly counts the number of times that track identity is preserved when difficult problems are encountered; hence it is not diluted by benign activity. This metric showed a dramatic *30% reduction in error* by the CAT system relative to the best-performing uniform background statistic system. In the population of 161 targets throughout the vignettes, this accounts for 19 tracks – nearly 12% of the population – which failed in the baseline system but were protected from track identity loss by the CAT system. In many concepts of employment, the protection of even one single track from identity loss is an important capability; any target could become high-value in a targeting or forensics application. This is perhaps the most important evidence that CAT is a viable approach.

5.1 Future Work

There are several opportunities for furthering this research effort. It remains to demonstrate that the adaptive background modeling results in Section 4.2.3 could approach the fidelity of the semi-automated method shown in Section 4.2.2. The key to this shortcoming: a proper region-splitting approach in the adaptive model is a ripe opportunity for future research.

Hospitability maps were briefly discussed in Section 2.3.3, where their significance to target state estimation was noted. This thesis has studied the generation of background models to derive detectability statistics solely for the sake of track scoring and maintenance. However, the same hyperspectral-derived background model could be used to aide in the formation of a hospitability map. This hospitability map could then be used in the target dynamics portion of the estimator. While doubly incorporating the background model – detectability and hospitability – requires a measure of caution, there is reason to believe that future study on such a system would be fruitful.

Another obvious extension of this effort would be to close the gap between the adaptive background model and the tracking testbed. In Section 4.2.3, it was noted that several simplifying assumptions were made for the sake of the iterative modeling approach and lack of operator input. In particular, the class hierarchy was simplified relative to that of the semi-automated approach in Section 4.2.2. Also, the NDVI and tree-index pre-processing technique which worked well in the semi-automated case was abandoned in favor of a classifier-only technique for the adaptive case. There may be merit in revisiting this decision, particularly if stable, scene-independent index thresholds could be determined for the separation of grass from tree canopies. Ultimately, these enhancements to the adaptive background model followed by its application in a tracking testbed would be a worthwhile endeavor.

In Section 2.4.4, it is observed that combining the HSI CAT system with an HSI FAT system in a common architecture is an enticing prospect. The HSI FAT performance shown in [54] coupled with the HSI CAT benefit shown in this thesis is highly synergistic. The recent availability of adaptive HSI instruments makes the performance gain even more likely.

Appendix A. Matlab Code

Here, Matlab® code is included for the entirety of the parametric experiment presented in Section 4.1.

Listing A.1: (costTest/runSims.m)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Simulation parameters
p = [];
p.time = 200;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Background parameters
p.pOcclusion = 0.01; %per-frame probability of an occlusion starting
% occlusion length is uniformly distributed between min and max:
p.minOcclusionDur = 1;
p.maxOcclusionDur = 20;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Target parameters
p.pTgtArrival   = 0.1; %per-frame probability of target arriving
p.pTgtDeparture = 0.005; %per-frame probability of target departing
p.kssMean = -5; %the mean kinematic cost of a healthy update
p.kssVar = 4; %variance in the above
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Measurement parameters
p.nominalClearPdTrue    = 0.95; %probability of a detection (clear)
p.nominalOccludedPdTrue = 0.05; %probability of a detection (occluded)
p.pFalseAlarm = 0.05; %per-frame unconditional probability of a false alarm
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tracker tuning parameters
p.Nconf = 5;  %prototypical num updates for confirm
p.Ndrop = 10; %prototypical window length for drop
p.Mdrop = 5;  %prototypical num misses for drop
confirmFactor = 5; %lower confirms sooner
dropFactor = 5; %lower drops sooner
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Tracker derived parameters
p.pssConf = p.kssMean+confirmFactor*sqrt(p.kssVar);
p.pssDrop = p.kssMean+dropFactor*sqrt(p.kssVar);
% tracker statistics
p.CAT = false;
p.CAT_Pd_clear = p.nominalClearPdTrue;
p.CAT_Pd_occluded = p.nominalOccludedPdTrue;
%p.CAT_Pd_occluded = 1e-2;
p.CAT_BetaNT_clear = 1e-2;
p.CAT_BetaNT_occluded = 1e-5;
p.unif_Pd = 0.97;
p.unif_BetaNT = 1e-2;
p.BetaFA_multiplier = 1e-2; %multiplied by pd

%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Which test to run?
% note: MC means Monte Carlo analysis
test = 0;

switch test
    case 0 % run one sim and plot it
        p.minOcclusionDur = 5;
        p.maxOcclusionDur = 20;
        p.unif_Pd = 0.8;
        sim = makeSim(p);
        plotSim(sim);
    case 1 % MC pd (occluded) vs betaFA multiplier sweep CAT
        tic;
        p.CAT = true;
```

```
        pd = 0.02:0.04:1;
        betaFA_mult = 10.^[1,0,-1,-2,-3,-4,-5,-6];
61      completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            for betaFAIter = 1:length(betaFA_mult)
66              p.CAT_Pd_occluded = pd(pdIter);
                p.BetaFA_multiplier = betaFA_mult(betaFAIter);
                mcRuns = 100;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
71              deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
76                  deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityByPd(1,pdIter,betaFAIter) = mean(purity);
                completenessByPd(1,pdIter,betaFAIter) = mean(completeness);
                deltaCardByPd(1,pdIter,betaFAIter) = mean(deltaCardinality);
81              purityByPd(2,pdIter,betaFAIter) = std(purity);
                completenessByPd(2,pdIter,betaFAIter) = std(completeness);
                deltaCardByPd(2,pdIter,betaFAIter) = std(deltaCardinality);
            end
        end
86      time = toc;
        figure;
        imagesc(squeeze(purityByPd(1,:,:)),[0.75,1.0]);
        set(gca,'XTick',1:length(betaFA_mult));
        set(gca,'XTickLabel',betaFA_mult);
91      xlabel('BetaFA multiplier');
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        ylabel('Pd occluded');
        colorbar;
96      title('pd_occluded / betaFA mult, purity, CAT');
        saveas(gcf,'1_pd_vs_betaFA_purity','fig');
        figure;
        imagesc(squeeze(completenessByPd(1,:,:)),[0.75,1.0]);
        set(gca,'XTick',1:length(betaFA_mult));
101     set(gca,'XTickLabel',betaFA_mult);
        xlabel('BetaFA multiplier');
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        ylabel('Pd occluded');
106     colorbar;
        title('pd_occluded / betaFA mult, completeness, CAT');
        saveas(gcf,'1_pd_vs_betaFA_completeness','fig');
        figure;
        imagesc(squeeze(deltaCardByPd(1,:,:)),[0,0.5]);
111     set(gca,'XTick',1:length(betaFA_mult));
        set(gca,'XTickLabel',betaFA_mult);
        xlabel('BetaFA multiplier');
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
116     ylabel('Pd occluded');
        colorbar;
        title('pd_occluded / betaFA mult, delta cardinality, CAT');
        saveas(gcf,'1_pd_vs_betaFA_deltaCard','fig');
        save 1_pd_vs_betaFA_CAT;
121 case 2 % MC pd vs betaFA multiplier sweep uniform
        tic;
        p.CAT = false;
        pd = 0.02:0.04:1;
```

```
        betaFA_mult = 10.^[1,0,-1,-2,-3,-4,-5,-6];
126     completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            for betaFAIter = 1:length(betaFA_mult)
131             p.unif_Pd = pd(pdIter);
                p.BetaFA_multiplier = betaFA_mult(betaFAIter);
                mcRuns = 100;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
136             deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
141                 deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityByPd(1,pdIter,betaFAIter) = mean(purity);
                completenessByPd(1,pdIter,betaFAIter) = mean(completeness);
                deltaCardByPd(1,pdIter,betaFAIter) = mean(deltaCardinality);
146             purityByPd(2,pdIter,betaFAIter) = std(purity);
                completenessByPd(2,pdIter,betaFAIter) = std(completeness);
                deltaCardByPd(2,pdIter,betaFAIter) = std(deltaCardinality);
            end
        end
151     time = toc;
        figure;
        imagesc(squeeze(purityByPd(1,:,:)),[0.75,1.0]);
        set(gca,'XTick',1:length(betaFA_mult));
        set(gca,'XTickLabel',betaFA_mult);
156     set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        title('pd / betaFA mult, purity, uniform');
        saveas(gcf,'2_pd_vs_betaFA_purity','fig');
161     figure;
        imagesc(squeeze(completenessByPd(1,:,:)),[0.75,1.0]);
        set(gca,'XTick',1:length(betaFA_mult));
        set(gca,'XTickLabel',betaFA_mult);
        set(gca,'YTick',1:length(pd));
166     set(gca,'YTickLabel',pd);
        colorbar;
        title('pd / betaFA mult, completeness, uniform');
        saveas(gcf,'2_pd_vs_betaFA_completeness','fig');
        figure;
171     imagesc(squeeze(deltaCardByPd(1,:,:)),[0,0.5]);
        set(gca,'XTick',1:length(betaFA_mult));
        set(gca,'XTickLabel',betaFA_mult);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
176     colorbar;
        title('pd / betaFA mult, delta cardinality, uniform');
        saveas(gcf,'2_pd_vs_betaFA_deltaCard','fig');
        save 2_pd_vs_betaFA_UNIF;
    case 3 % MC Mdrop Ndrop sweep UNIFORM
181     tic;
        p.CAT = false;
        Mvals = [2,3,4,5,7,9];
        Nvals = [10,15,20,25,30];
        for Miter = 1:length(Mvals)
186         for Niter = 1:length(Nvals)
                p.Ndrop = Nvals(Niter);
                p.Mdrop = Mvals(Miter);
                mcRuns = 200;
                completeness     = zeros(1,mcRuns);
```

```
                purity           = zeros(1,mcRuns);
                deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
                    deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityAll(Miter,Niter) = mean(purity);
                completenessAll(Miter,Niter) = mean(completeness);
                deltaCardAll(Miter,Niter) = mean(deltaCardinality);
            end
        end
        time = toc
        figure;
        imagesc(purityAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, purity, uniform');
        saveas(gcf,'3_Mdrop_Ndrop_purity','fig');
        figure;
        imagesc(completenessAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, completeness, uniform');
        saveas(gcf,'3_Mdrop_Ndrop_completeness','fig');
        figure;
        imagesc(deltaCardAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, delta cardinality, uniform');
        saveas(gcf,'3_Mdrop_Ndrop_deltaCard','fig');
        save 3_Mdrop_Ndrop_UNIF;
    case 4 % MC Mdrop Ndrop sweep CAT
        tic;
        p.CAT = true;
        Mvals = [2,3,4,5,7,9];
        Nvals = [10,15,20,25,30];
        for Miter = 1:length(Mvals)
            for Niter = 1:length(Nvals)
                p.Ndrop = Nvals(Niter);
                p.Mdrop = Mvals(Miter);
                mcRuns = 200;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
                deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
                    deltaCardinality(mcIter) = sim.deltaCardinality;
```

```
                end
                purityAll(Miter,Niter) = mean(purity);
                completenessAll(Miter,Niter) = mean(completeness);
                deltaCardAll(Miter,Niter) = mean(deltaCardinality);
            end
        end
        time = toc
        figure;
        imagesc(purityAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, purity, CAT');
        saveas(gcf,'4_Mdrop_Ndrop_purity','fig');
        figure;
        imagesc(completenessAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, completeness, CAT');
        saveas(gcf,'4_Mdrop_Ndrop_completeness','fig');
        figure;
        imagesc(deltaCardAll);
        set(gca,'XTick',1:length(Nvals));
        set(gca,'XTickLabel',Nvals);
        xlabel('Ndrop');
        set(gca,'YTick',1:length(Mvals));
        set(gca,'YTickLabel',Mvals);
        ylabel('Mdrop');
        colorbar;
        title('Mdrop / Ndrop, delta cardinality, CAT');
        saveas(gcf,'4_Mdrop_Ndrop_deltaCard','fig');
        save 4_Mdrop_Ndrop_CAT;
    case 5 %Uniform Pd sweep, all metrics
        tic;
        p.CAT = false;
        pd = 0.02:0.04:1;
        completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            p.unif_Pd = pd(pdIter);
            mcRuns = 200;
            completeness     = zeros(1,mcRuns);
            purity           = zeros(1,mcRuns);
            deltaCardinality = zeros(1,mcRuns);
            for mcIter = 1:mcRuns
                sim = makeSim(p);
                completeness(mcIter) = sim.meanCompleteness;
                purity(mcIter) = sim.purity;
                deltaCardinality(mcIter) = sim.deltaCardinality;
            end
            purityByPd(1,pdIter) = mean(purity);
            completenessByPd(1,pdIter) = mean(completeness);
            deltaCardByPd(1,pdIter) = mean(deltaCardinality);
            purityByPd(2,pdIter) = std(purity);
            completenessByPd(2,pdIter) = std(completeness);
            deltaCardByPd(2,pdIter) = std(deltaCardinality);
```

```
        end
        time = toc;
        figure;
        h=[];
        legends = {};
        hold on;
        h(end+1)=plot(pd,purityByPd(1,:),'bo-');
        plot(pd,purityByPd(1,:)-purityByPd(2,:),'b--');
        plot(pd,purityByPd(1,:)+purityByPd(2,:),'b--');
        legends{end+1} = 'Purity';
        h(end+1)=plot(pd,completenessByPd(1,:),'gd-');
        plot(pd,completenessByPd(1,:)-completenessByPd(2,:),'g--');
        plot(pd,completenessByPd(1,:)+completenessByPd(2,:),'g--');
        legends{end+1} = 'Completeness';
        h(end+1)=plot(pd,deltaCardByPd(1,:),'ms-');
        plot(pd,deltaCardByPd(1,:)-deltaCardByPd(2,:),'m--');
        plot(pd,deltaCardByPd(1,:)+deltaCardByPd(2,:),'m--');
        legends{end+1} = 'Delta Cardinality';
        legend(h,legends);
        xlabel('Pd');
        saveas(gcf,'5_Pd_sweep_uniform','fig');
        save 5_Pd_sweep_uniform;
    case 6 %Uniform Pd sweep, all metrics, logarithmic end test
        tic;
        p.CAT = false;
        pd = 1-10.^-[1:10];
        completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            p.unif_Pd = pd(pdIter);
            mcRuns = 200;
            completeness     = zeros(1,mcRuns);
            purity           = zeros(1,mcRuns);
            deltaCardinality = zeros(1,mcRuns);
            for mcIter = 1:mcRuns
                sim = makeSim(p);
                completeness(mcIter) = sim.meanCompleteness;
                purity(mcIter) = sim.purity;
                deltaCardinality(mcIter) = sim.deltaCardinality;
            end
            purityByPd(1,pdIter) = mean(purity);
            completenessByPd(1,pdIter) = mean(completeness);
            deltaCardByPd(1,pdIter) = mean(deltaCardinality);
            purityByPd(2,pdIter) = std(purity);
            completenessByPd(2,pdIter) = std(completeness);
            deltaCardByPd(2,pdIter) = std(deltaCardinality);
        end
        time = toc;
        figure;
        h=[];
        legends = {};
        hold on;
        h(end+1)=plot(purityByPd(1,:),'bo-');
        plot(purityByPd(1,:)-purityByPd(2,:),'b--');
        plot(purityByPd(1,:)+purityByPd(2,:),'b--');
        legends{end+1} = 'Purity';
        h(end+1)=plot(completenessByPd(1,:),'gd-');
        plot(completenessByPd(1,:)-completenessByPd(2,:),'g--');
        plot(completenessByPd(1,:)+completenessByPd(2,:),'g--');
        legends{end+1} = 'Completeness';
        h(end+1)=plot(deltaCardByPd(1,:),'ms-');
        plot(deltaCardByPd(1,:)-deltaCardByPd(2,:),'m--');
        plot(deltaCardByPd(1,:)+deltaCardByPd(2,:),'m--');
        legends{end+1} = 'Delta Cardinality';
        legend(h,legends);
```

```
        xlabel('Pd');
        set(gca,'XTick',1:length(pd));
        set(gca,'XTickLabel',pd);
        saveas(gcf,'6_Pd_sweep_uniform_logend','fig');
        save 6_Pd_sweep_uniform_logend;
    case 7 %CAT Pd (clear) sweep, all metrics
        tic;
        p.CAT = true;
        pd = 0.02:0.04:1;
        completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            p.CAT_Pd_clear = pd(pdIter);
            mcRuns = 200;
            completeness     = zeros(1,mcRuns);
            purity           = zeros(1,mcRuns);
            deltaCardinality = zeros(1,mcRuns);
            for mcIter = 1:mcRuns
                sim = makeSim(p);
                completeness(mcIter) = sim.meanCompleteness;
                purity(mcIter) = sim.purity;
                deltaCardinality(mcIter) = sim.deltaCardinality;
            end
            purityByPd(1,pdIter) = mean(purity);
            completenessByPd(1,pdIter) = mean(completeness);
            deltaCardByPd(1,pdIter) = mean(deltaCardinality);
            purityByPd(2,pdIter) = std(purity);
            completenessByPd(2,pdIter) = std(completeness);
            deltaCardByPd(2,pdIter) = std(deltaCardinality);
        end
        time = toc;
        figure;
        h=[];
        legends = {};
        hold on;
        h(end+1)=plot(pd,purityByPd(1,:),'bo-');
        plot(pd,purityByPd(1,:)-purityByPd(2,:),'b--');
        plot(pd,purityByPd(1,:)+purityByPd(2,:),'b--');
        legends{end+1} = 'Purity';
        h(end+1)=plot(pd,completenessByPd(1,:),'gd-');
        plot(pd,completenessByPd(1,:)-completenessByPd(2,:),'g--');
        plot(pd,completenessByPd(1,:)+completenessByPd(2,:),'g--');
        legends{end+1} = 'Completeness';
        h(end+1)=plot(pd,deltaCardByPd(1,:),'ms-');
        plot(pd,deltaCardByPd(1,:)-deltaCardByPd(2,:),'m--');
        plot(pd,deltaCardByPd(1,:)+deltaCardByPd(2,:),'m--');
        legends{end+1} = 'Delta Cardinality';
        legend(h,legends);
        xlabel('Pd (clear)');
        saveas(gcf,'7_Pd_clear_sweep_CAT','fig');
        save 7_Pd_clear_sweep_CAT;
    case 8 %CAT Pd (occluded) sweep, all metrics
        tic;
        p.CAT = true;
        pd = 0.02:0.04:1;
        completenessByPd = zeros(2,length(pd));
        purityByPd = zeros(2,length(pd));
        deltaCardByPd = zeros(2,length(pd));
        for pdIter = 1:length(pd)
            p.CAT_Pd_occluded = pd(pdIter);
            mcRuns = 200;
            completeness     = zeros(1,mcRuns);
            purity           = zeros(1,mcRuns);
            deltaCardinality = zeros(1,mcRuns);
            for mcIter = 1:mcRuns
```

```
                sim = makeSim(p);
                completeness(mcIter) = sim.meanCompleteness;
                purity(mcIter) = sim.purity;
                deltaCardinality(mcIter) = sim.deltaCardinality;
            end
            purityByPd(1,pdIter) = mean(purity);
            completenessByPd(1,pdIter) = mean(completeness);
            deltaCardByPd(1,pdIter) = mean(deltaCardinality);
            purityByPd(2,pdIter) = std(purity);
            completenessByPd(2,pdIter) = std(completeness);
            deltaCardByPd(2,pdIter) = std(deltaCardinality);
        end
        time = toc;
        figure;
        h=[];
        legends = {};
        hold on;
        h(end+1)=plot(pd,purityByPd(1,:),'bo-');
        plot(pd,purityByPd(1,:)-purityByPd(2,:),'b--');
        plot(pd,purityByPd(1,:)+purityByPd(2,:),'b--');
        legends{end+1} = 'Purity';
        h(end+1)=plot(pd,completenessByPd(1,:),'gd-');
        plot(pd,completenessByPd(1,:)-completenessByPd(2,:),'g--');
        plot(pd,completenessByPd(1,:)+completenessByPd(2,:),'g--');
        legends{end+1} = 'Completeness';
        h(end+1)=plot(pd,deltaCardByPd(1,:),'ms-');
        plot(pd,deltaCardByPd(1,:)-deltaCardByPd(2,:),'m--');
        plot(pd,deltaCardByPd(1,:)+deltaCardByPd(2,:),'m--');
        legends{end+1} = 'Delta Cardinality';
        legend(h,legends);
        xlabel('Pd (occluded)');
        saveas(gcf,'8_Pd_occluded_sweep_CAT','fig');
        save 8_Pd_occluded_sweep_CAT;
    case 9 %CAT Pd (occluded) vs p.nominalOccludedPdTrue
        tic;
        p.CAT = true;
        %consider stopping false alarms?
        pd = 0.01:0.01:0.5;
        pm = 0.00:0.05:0.5;
        for pdIter = 1:length(pd)
            for pmIter = 1:length(pm)
                p.CAT_Pd_occluded = pd(pdIter);
                p.nominalOccludedPdTrue = pm(pmIter);
                mcRuns = 200;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
                deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
                    deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityByPd(pdIter,pmIter) = mean(purity);
                completenessByPd(pdIter,pmIter) = mean(completeness);
                deltaCardByPd(pdIter,pmIter) = mean(deltaCardinality);
            end
        end
        time = toc;
        figure;
        imagesc(purityByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
```

```
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, purity, CAT');
        saveas(gcf,'9_pd_vs_pm_purity','fig');
        figure;
        imagesc(completenessByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, completeness, CAT');
        saveas(gcf,'9_pd_vs_pm_completeness','fig');
        figure;
        imagesc(deltaCardByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, delta cardinality, CAT');
        saveas(gcf,'9_pd_vs_pm_deltacard','fig');
        save 9_Pd_Pm_CAT;
    case 10 %CAT Pd (occluded) vs p.nominalOccludedPdTrue NO FALSE ALARMS
        tic;
        p.CAT = true;
        p.pFalseAlarm = 0;
        pd = 0.01:0.01:0.5;
        pm = 0.00:0.05:0.5;
        clear purityByPd completenessByPd deltaCard;
        for pdIter = 1:length(pd)
            for pmIter = 1:length(pm)
                p.CAT_Pd_occluded = pd(pdIter);
                p.nominalOccludedPdTrue = pm(pmIter);
                mcRuns = 200;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
                deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
                    deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityByPd(pdIter,pmIter) = mean(purity);
                completenessByPd(pdIter,pmIter) = mean(completeness);
                deltaCardByPd(pdIter,pmIter) = mean(deltaCardinality);
            end
        end
        time = toc;
        figure;
        imagesc(purityByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, purity, CAT');
        saveas(gcf,'10_pd_vs_pm_purity_noFA','fig');
        figure;
```

```
        imagesc(completenessByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, completeness, CAT');
        saveas(gcf,'10_pd_vs_pm_completeness_noFA','fig');
        figure;
        imagesc(deltaCardByPd);
        set(gca,'XTick',1:length(pm));
        set(gca,'XTickLabel',pm);
        set(gca,'YTick',1:length(pd));
        set(gca,'YTickLabel',pd);
        colorbar;
        xlabel('p measurement during occlusion');
        ylabel('Pd occluded');
        title('pd vs pm, delta cardinality, CAT');
        saveas(gcf,'10_pd_vs_pm_deltacard_noFA','fig');
        save 10_Pd_Pm_CAT_noFA;
    case 11 %vary the occlusion frequency / duration, unif
        tic;
        p.CAT = false;
        occlusionDurMinMax = ...
            [1, 2,  5, 7,  10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35
            5,  7, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40];
        pOcclusion = 0:0.01:0.2;
        clear purityByPd completenessByPd deltaCard;
        for occlusionDurIter = 1:size(occlusionDurMinMax,2)
            for pOcclusionIter = 1:length(pOcclusion)
                p.minOcclusionDur = occlusionDurMinMax(1,occlusionDurIter);
                p.maxOcclusionDur = occlusionDurMinMax(2,occlusionDurIter);
                p.pOcclusion = pOcclusion(pOcclusionIter);
                mcRuns = 200;
                completeness     = zeros(1,mcRuns);
                purity           = zeros(1,mcRuns);
                deltaCardinality = zeros(1,mcRuns);
                for mcIter = 1:mcRuns
                    sim = makeSim(p);
                    completeness(mcIter) = sim.meanCompleteness;
                    purity(mcIter) = sim.purity;
                    deltaCardinality(mcIter) = sim.deltaCardinality;
                end
                purityByPd(occlusionDurIter,pOcclusionIter) = mean(purity);
                completenessByPd(occlusionDurIter,pOcclusionIter) = mean(...
                    completeness);
                deltaCardByPd(occlusionDurIter,pOcclusionIter) = mean(...
                    deltaCardinality);
            end
        end
        time = toc;
        figure;
        imagesc(purityByPd,[0,1]);
        set(gca,'XTick',1:length(pOcclusion));
        set(gca,'XTickLabel',pOcclusion);
        set(gca,'YTick',1:size(occlusionDurMinMax,2));
        set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
        colorbar;
        xlabel('P occlusion per frame');
        ylabel('Occlusion Duration min/max');
        title('pOcclusion vs occlusionDur, purity, Unif');
        saveas(gcf,'11_pd_vs_pm_purity_unif','fig');
        figure;
        imagesc(completenessByPd,[0,1]);
```

```
651         set(gca,'XTick',1:length(pOcclusion));
            set(gca,'XTickLabel',pOcclusion);
            set(gca,'YTick',1:size(occlusionDurMinMax,2));
            set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
            colorbar;
656         xlabel('P occlusion per frame');
            ylabel('Occlusion Duration min/max');
            title('pOcclusion vs occlusionDur, completeness, Unif');
            saveas(gcf,'11_pd_vs_pm_completeness_unif','fig');
            figure;
661         imagesc(deltaCardByPd,[0,1]);
            set(gca,'XTick',1:length(pOcclusion));
            set(gca,'XTickLabel',pOcclusion);
            set(gca,'YTick',1:size(occlusionDurMinMax,2));
            set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
666         colorbar;
            xlabel('P occlusion per frame');
            ylabel('Occlusion Duration min/max');
            title('pOcclusion vs occlusionDur, delta cardinality, Unif');
            saveas(gcf,'11_pd_vs_pm_deltacard_unif','fig');
671         save 11_Pd_Pm_unif;
        case 12 %vary the occlusion frequency / duration, CAT
            tic;
            p.CAT = true;
            occlusionDurMinMax = ...
676             [1, 2,  5, 7,  10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35
                5,  7, 10, 13, 15, 18, 20, 23, 25, 28, 30, 33, 35, 38, 40];
            pOcclusion = 0:0.01:0.2;
            clear purityByPd completenessByPd deltaCard;
            for occlusionDurIter = 1:size(occlusionDurMinMax,2)
681             for pOcclusionIter = 1:length(pOcclusion)
                    p.minOcclusionDur = occlusionDurMinMax(1,occlusionDurIter);
                    p.maxOcclusionDur = occlusionDurMinMax(2,occlusionDurIter);
                    p.pOcclusion = pOcclusion(pOcclusionIter);
                    mcRuns = 200;
686                 completeness     = zeros(1,mcRuns);
                    purity           = zeros(1,mcRuns);
                    deltaCardinality = zeros(1,mcRuns);
                    for mcIter = 1:mcRuns
                        sim = makeSim(p);
691                     completeness(mcIter) = sim.meanCompleteness;
                        purity(mcIter) = sim.purity;
                        deltaCardinality(mcIter) = sim.deltaCardinality;
                    end
                    purityByPd(occlusionDurIter,pOcclusionIter) = mean(purity);
696                 completenessByPd(occlusionDurIter,pOcclusionIter) = mean(...
                        completeness);
                    deltaCardByPd(occlusionDurIter,pOcclusionIter) = mean(...
                        deltaCardinality);
                end
            end
            time = toc;
701         figure;
            imagesc(purityByPd,[0,1]);
            set(gca,'XTick',1:length(pOcclusion));
            set(gca,'XTickLabel',pOcclusion);
            set(gca,'YTick',1:size(occlusionDurMinMax,2));
706         set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
            colorbar;
            xlabel('P occlusion per frame');
            ylabel('Occlusion Duration min/max');
            title('pOcclusion vs occlusionDur, purity, CAT');
711         saveas(gcf,'12_pd_vs_pm_purity_CAT','fig');
            figure;
            imagesc(completenessByPd,[0,1]);
            set(gca,'XTick',1:length(pOcclusion));
```

```
            set(gca,'XTickLabel',pOcclusion);
716         set(gca,'YTick',1:size(occlusionDurMinMax,2));
            set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
            colorbar;
            xlabel('P occlusion per frame');
            ylabel('Occlusion Duration min/max');
721         title('pOcclusion vs occlusionDur, completeness, CAT');
            saveas(gcf,'12_pd_vs_pm_completeness_CAT','fig');
            figure;
            imagesc(deltaCardByPd,[0,1]);
            set(gca,'XTick',1:length(pOcclusion));
726         set(gca,'XTickLabel',pOcclusion);
            set(gca,'YTick',1:size(occlusionDurMinMax,2));
            set(gca,'YTickLabel',sprintf('%d,%d|',occlusionDurMinMax));
            colorbar;
            xlabel('P occlusion per frame');
731         ylabel('Occlusion Duration min/max');
            title('pOcclusion vs occlusionDur, delta cardinality, CAT');
            saveas(gcf,'12_pd_vs_pm_deltacard_CAT','fig');
            save 12_Pd_Pm_CAT;
    end
```

Listing A.2: (costTest/makeSim.m)

```
    function sim = makeSim(p)
    % Function sim = makeSim(p)
    %       Create one run through the simulator.
  4 %
    % Input:
    %   p - parameter structure for the simulation with fields:
    %
    % Output:
  9 %   sim - output structure for the simulation
    %
    % Andrew C. Rice, andrewcrice@gmail.com

    % Preparation
 14 bg   = makeBackground(p.time, p.pOcclusion, p.minOcclusionDur, p.maxOcclusionDur);
    tgt  = makeTarget(p.time, p.pTgtArrival, p.pTgtDeparture);
    pDetectTrue = ones(1,p.time);
    pDetectTrue(bg)  = p.nominalClearPdTrue;
    pDetectTrue(~bg) = p.nominalOccludedPdTrue;
 19 meases = makeMeasurements(tgt, pDetectTrue, p.pFalseAlarm);

    % Track costing
    if p.CAT
        pDetectAssumed = ones(1,p.time);
 24     pDetectAssumed(bg) = p.CAT_Pd_clear;
        pDetectAssumed(~bg) = p.CAT_Pd_occluded;
        betaNTassumed  = ones(1,p.time);
        betaNTassumed(bg) = p.CAT_BetaNT_clear;
        betaNTassumed(~bg) = p.CAT_BetaNT_occluded;
 29 else
        pDetectAssumed = ones(1,p.time) * p.unif_Pd;
        betaNTassumed = ones(1,p.time) * p.unif_BetaNT;
    end
    betaFAassumed = pDetectAssumed * p.BetaFA_multiplier;
 34 costInstant = trackInstantaneousCost(meases, pDetectAssumed, ...
        betaFAassumed, p.kssMean, p.kssVar);
    costCumulative = NaN(1,length(costInstant));
    costWindowed = costCumulative;

 39 % Track maintenance
    allOrigins = [];        %frame indeces of track origins
    allConfirmations = [];  %frame indeces of track confirmations
    allDrops = [];          %frame indeces of track drops
```

```
trk = zeros(1,p.time);  %vector which is zero when no mature track exists,
                        %and a monatonically increasing track ID when a
                        %mature track does exist.
trkID = 0;
maintenanceFrame = 1;
while maintenanceFrame <= p.time
    [origin,confirmed] = whenConfirmed(meases, costInstant, pDetectAssumed,...
        p.Nconf, p.pssConf, betaNTassumed, betaFAassumed, ...
        maintenanceFrame);
    if confirmed > 0
        costCumulative(origin:end) = cumsum(costInstant(origin:end));
        thisCostWindowed = trackWindowedCost(costInstant, p.Ndrop, origin);
        costWindowed(origin:end) = thisCostWindowed(origin:end);
        dropped = whenDropped(thisCostWindowed, pDetectAssumed, p.Mdrop, ...
            p.Ndrop, p.pssDrop, betaFAassumed, origin, confirmed+1);
        allOrigins(end+1) = origin;
        allConfirmations(end+1) = confirmed;
        allDrops(end+1) = dropped;
        trkID = trkID+1;
        if dropped == 0
            %ran out of frames before dropping
            trk(confirmed:end) = trkID;
            break;
        else
            trk(confirmed:dropped) = trkID;
            maintenanceFrame = dropped+1;
            costCumulative(maintenanceFrame:end) = NaN;
            costWindowed(maintenanceFrame:end) = NaN;
        end
    else
        %no (additional) confirmations to report
        break;
    end
end

% Metrics
if trkID == 0
    %never formed a track, special case of metrics
    sim.meanCompleteness = 0;
    sim.purity = 0;
    sim.deltaCardinality = 1; %represents worst possible
else
    sim.meanCompleteness = sum(tgt .* (trk > 0)) / sum(tgt);
    trkIDwhenTracked = trk(trk~=0);
    sim.purity = sum(trk==mode(trkIDwhenTracked)) / length(trkIDwhenTracked);
    sim.deltaCardinality = sum(abs(tgt ~= (trk > 0))) / p.time;
end
if any(isnan([sim.meanCompleteness, sim.purity, sim.deltaCardinality]))
    warning;
end

% Output
sim.tgt = tgt;
sim.meases = meases;
sim.trk = trk;
sim.confirmed = allConfirmations;
sim.dropped = allDrops;
sim.costInstant = costInstant;
sim.costCumulative = costCumulative;
sim.costWindowed = costWindowed;
sim.Pd = pDetectTrue;
```

Listing A.3: (costTest/makeBackground.m)

```
function bg = makeBackground(time, pOcclusion, minOcclusionDur, ...
    maxOcclusionDur)
```

```
% Function bg = makeBackground(time, pOcclusion, minOcclusionDur, ...
%                               maxOcclusionDur)
%       Compute a random background which is characterized by the presence
%       or absence of occlusions. Any number of occlusions may occur.
% Input:
%   time - number of frames for the background lifetime.
%   pOcclusion - per-frame probability of an occlusion occuring
%   minOcclusionDur - minimum length of an occlusion
%   maxOcclusionDur - maximum length of an occlusion
%
% Output:
%   bg - [1xtime] vector of background truth (1=unoccluded, 0=occluded)
%
% Andrew C. Rice, andrewcrice@gmail.com

bg = true(1,time);
now = 0;
while true
    now = now + randGeometric(pOcclusion);
    if now > time; break; end;
    duration = randi([minOcclusionDur, maxOcclusionDur]);
    duration = min(duration, time-now);
    bg(now:now+duration) = false;
end
```

Listing A.4: (costTest/makeTarget.m)

```
function tgt = makeTarget(time, pArrival, pDeparture)
% Function tgt = makeTarget(time, pArrival, pDeparture)
%       Compute a random target arrival and departure.
% Input:
%   time - number of frames for the target lifetime.
%   pArrival - per-frame probability of a target arriving
%   pDeparture - per-frame probability of a target departing
%
% Output:
%   tgt - [1xtime] vector of target truth (1=present, 0=absent)
%
% Andrew C. Rice, andrewcrice@gmail.com

tgt = false(1,time);
arrive = randGeometric(pArrival);
if arrive < time
    depart = randGeometric(pDeparture);
    if arrive + depart < time
        tgt(arrive:arrive+depart) = true;
    else
        %never departed
        tgt(arrive:end) = true;
    end
else
    %never arrived
end
```

Listing A.5: (costTest/makeMeasurements.m)

```
function meases = makeMeasurements(tgt, pDetect, pFalseAlarm)
% Function meases = makeMeasurements(tgt, pDetect, pFalseAlarm)
%       Compute a measurement sequence for the target based upon its
%       presence and probability of detection at each frame.
% Input:
%   tgt - [1xn] target presence/absence vector
%   pDetect - [1xn] probability of detection vector
%   pFalseAlarm - per-frame probability of a false alarm
%
```

```
% Output:
%   meases - [1xn] vector of measurements (1=measurement, 0=no measurement)
%
% Andrew C. Rice, andrewcrice@gmail.com

meases = false(size(tgt));

% create false alarm measurements
meases(rand(1,length(meases)) < pFalseAlarm) = true;

% create target measurements according to Pd
meases(rand(1,length(meases)) < (tgt .* pDetect)) = true;
```

Listing A.6: (costTest/plotSim.m)

```
function plotSim(sim)
% Function confirmed = plotsim(sim)
%       Plot the output from the simulation run.
% Input:
%   sim - structure of simulation output
%
% Output:
%   n/a
%
% Andrew C. Rice, andrewcrice@gmail.com

% time frames
t = 1:length(sim.tgt);

% plot prep
figure(100);
clf;
hold on; h = []; legends = {};
minT = 1; % first frame to plot
axMain = gca; % the main plot area for the cost
marg = 0.15; %margin
doTexts = true;
doLegend = true;

%plot pd
bump = 0.03;  % a little space between the axes
axPd = axes('Position', ...
    [marg,0.8*(1-2*marg)+marg+bump,1-2*marg,0.15*(1-2*marg)-bump], ...
    'Color','none');
hold on;
plot(t, sim.Pd, 'k', 'LineWidth', 2);
ylabel('$P_\mathrm{D}$','Interpreter','latex');
axis([minT,max(t),0,1]);

%plot detections, truth
axDet = axes('Position', ...
    [marg,0.7*(1-2*marg)+marg+bump,1-2*marg,0.1], ...
    'Color','none');
hold on;
h(end+1) = plot(t(sim.tgt), 0.02*ones(1,sum(sim.tgt)),'k.','MarkerSize',7);
legends{end+1} = 'Target';
hit = find(sim.meases .* sim.tgt);
if ~isempty(hit)
    h(end+1) = plot(t(hit),  0.0*ones(1,length(hit)),'g.','MarkerSize',7);
    legends{end+1} = 'Measurement';
end
falseAlarm = find(sim.meases .* ~sim.tgt);
if ~isempty(falseAlarm)
    h(end+1) = plot(t(falseAlarm),  0.0*ones(1,length(falseAlarm)), ...
        'rx','MarkerSize',7,'LineWidth',1);
    legends{end+1} = 'False Alarm';
```

```
    end
    misses = find(~sim.meases .* sim.tgt);
    if ~isempty(misses)
        h(end+1) = plot(t(misses), 0.0*ones(1,length(misses)), ...
            'r.','MarkerSize',7,'LineWidth',1);
        legends{end+1} = 'Miss';
    end
    axis([minT,max(t),0,0.1]);
    axis off;

    %plot windowed cost
    axes(axMain);
    h(end+1) = plot(t, sim.costWindowed, 'm-s', 'LineWidth',1,'MarkerSize',7);
    legends{end+1} = '$\bar{C}$';
    axisCost = axis;
    axisCost(1:2) = [minT,max(t)];
    axis(axisCost);
    %plot confirmed
    for confirmed = sim.confirmed
        if confirmed > 0
            plot(confirmed*[1,1], [axisCost(3),axisCost(4)],'g--','LineWidth',1);
            if doTexts
                text(confirmed, axisCost(3)+2*(axisCost(4)-axisCost(3))/3, ...
                    '$\Longleftarrow \mathrm{T}_\mathrm{conf} \ge C$', ...
                    'Interpreter','latex');
            end
        end
    end
    %plot dropped
    for dropped = sim.dropped
        if dropped > 0
            plot(dropped*[1,1], [axisCost(3),axisCost(4)],'r--','LineWidth',1);
            if doTexts
                text(dropped, axisCost(3)+(axisCost(4)-axisCost(3))/3, ...
                    '$\Longleftarrow \bar{C} \ge \mathrm{T}_\mathrm{drop}$', ...
                    'Interpreter','latex');
            end
        end
    end
    %finish the main plot
    ylabel('cost','Interpreter','latex');
    xlabel('time','Interpreter','latex');
    set(axMain,'Position',[marg,marg,1-2*marg,0.7*(1-2*marg)],'Color','none');

    %create legend box
    if doLegend
        axes(axMain);
        lh = legend(h, legends);
        set(lh,'Interpreter','latex','Position',[0.7,0.25,0.15,0.25]);
    end

    %print plot
    set(gcf,'Position',[680 658 560*3 280*1.5]); %bigger
    set(gcf,'PaperPositionMode','auto');
    print('-depsc','winDropPlot.eps');
```

Listing A.7: (costTest/randGeometric.m)

```
    function g = randGeometric(p,n)
    % Function g = randGeometric(p,n)
    %       Compute random draws from the discrete geometric distribution.
    % Input:
    %   p - probability of success for independent Bernoulli trial
    %       (default 0.5)
    %   n - number of draws to return from the geometric discrete distribution
    %
```

```
% Output:
%   g - [1xn] vector of random draws from the geometric distribution
%
% Cited:
% Simulating Discrete (Geometric, Poisson and Zero-Inflated Poisson,
% Negative Binomial and Zero-Inflated Negative Binomial) Random Variables
% from http://www.ats.ucla.edu/stat/stata/code/discrete_rv_v2.htm
% (accessed 2011-04-05)
%
% Andrew C. Rice, andrewcrice@gmail.com

if nargin < 1; p = 0.5; end
if nargin < 2; n = 1; end
if p > 1; p = 1; end
if p < eps
    g = Inf;
    return
end
u = rand(1,n);
g = floor(log(u)/log(1-p))+1;
```

Listing A.8: (costTest/trackInstantaneousCost.m)

```
function costInstant = trackInstantaneousCost(meases, pDetectAssumed, ...
    betaFAassumed, kssMean, kssVar)
% Function costInstant = trackInstantaneousCost(meases, pDetectAssumed, ...
%    betaFAassumed, kssMean, kssVar)
%       Compute the per-frame instantaneous cost of the track.
% Input:
%   meases - [1xn] measurement presence/absence vector
%   pDetectAssumed - [1xn] the assumed probability of detection vector
%   betaFAassumed - [1xn] the assumed false alarm rate
%   kssMean - the steady-state kinematic distance^2
%   kssVar  - variance of the steady-state kinematic distance^2
%
% Output:
%   costInstant - [1xn] vector of the instantaneous cost at each frame
%
% Andrew C. Rice, andrewcrice@gmail.com

n = length(meases);
costInstant = zeros(1,n);

% the kinmatic portion of the cost must be randomly drawn since the actual
% tracker is missing
noise = randn(1,n);

costHit = -log(pDetectAssumed./betaFAassumed) + (kssMean+sqrt(kssVar).*noise);
costMiss = -log(1-pDetectAssumed);

% the instantaneous cost
costInstant(meases) = costHit(meases);
costInstant(~meases) = costMiss(~meases);
```

Listing A.9: (costTest/trackWindowedCost.m)

```
function costWindowed = trackWindowedCost(costInstant, winLen, origin)
% Function costWindowed = trackWindowedCost(costInstant, winLen)
%       Compute the per-frame windowed cost of the track. The window
%       undergoes a filling period initially, such that early costs will
%       perhaps include less than winLen elements. The window then slides
%       with time.
% Input:
%   costInstant - [1xn] instantaneous cost vector
%   winLen  - window length
```

```
%   origin  - frame index of the origin of the track
%
% Output:
%   costWindowed - [1xn] vector of the windowed cost at each frame. Nan
%                  prior to origin
%
% Andrew C. Rice, andrewcrice@gmail.com

costWindowed = zeros(size(costInstant));
for winEnd = 1:length(costInstant)
    if winEnd < origin
        costWindowed(winEnd) = NaN;
    else
        winStart = max(origin,winEnd-winLen+1);
        costWindowed(winEnd) = sum(costInstant(winStart:winEnd));
    end
end
```

Listing A.10: (costTest/whenConfirmed.m)

```
function [origin,confirmed] = whenConfirmed(meases, cost, Pd, Nconf, ...
    pssConf, betaNT, betaFA, beginAt)
% Function [origin,confirmed] = whenConfirmed(meases, cost, Pd, Nconf, ...
%                                   pssConf, betaNT, betaFA, beginAt)
%       Apply the track confirmation logic to determine when (if) the track
%       reaches confirmation status. Single frame initiation is used such
%       that each measurement is a candidate for becoming the origin of the
%       track. The chosen origin is the measurement which results in the
%       earliest confirmation. This mimics a simple tracking system.
% Input:
%   meases - [1xn] boolean measurement vector
%   cost - [1xn] instantaneous cost vector (NOT a windowed cost)
%   Pd  - [1xn] probability of detection (assumed)
%   Nconf - number of updates for the prototype benchmark track
%   pssConf - steady state kinematic p(z|x)
%   betaNT - [1xn] assumed new track density
%   betaFA - [1xn] assumed false alarm density
%   beginAt - Index of first frame to consider the origin.
%
% Output:
%   origin - index of the selected origin frame in [1,n], or 0 if never
%            confirmed
%   confirmed - index of the frame of confirmation. In [1,n], or 0 if never
%               confirmed.
%
% Andrew C. Rice, andrewcrice@gmail.com

candidateOrigins = find(meases);
bestConfirmed = Inf;
bestOrigin = 0;
% the following for loop and while loop test potential origin and
% confirmation frames. Note that a later origin may sometimes result in a
% quicker confirmation depending on the statistics involved. Hence the
% exhaustive search. In the real tracker, this mimics competing tracks
% within the MHT.
for candidateOrigin = candidateOrigins
    if candidateOrigin < beginAt
        %too early in sequence
        continue
    end
    thisCost = cost;
    thisCost(1:candidateOrigin-1) = 0;
    thisCostCum = cumsum(thisCost);
    frameIter = candidateOrigin;
    while frameIter < length(cost)
        thisMinPd = min(Pd(candidateOrigin:frameIter));
```

```
            thisThresh = -log(betaNT(candidateOrigin)/betaFA(candidateOrigin)) ...
                - Nconf * log(thisMinPd*pssConf/betaFA(frameIter));
            if thisCostCum(frameIter) < thisThresh
                % a confirmation, test to see if it's the best so far
                if frameIter < bestConfirmed
                    bestConfirmed = frameIter;
                    bestOrigin = candidateOrigin;
                end
                break; %end the while loop for this candidateOrigin
            end
            frameIter = frameIter + 1;
        end
    end
    if bestConfirmed == Inf
        % never confirmed
        confirmed = 0;
        origin = 0;
    else
        confirmed = bestConfirmed;
        origin = bestOrigin;
    end
```

Listing A.11: (costTest/whenDropped.m)

```
function dropped = whenDropped(cost, Pd, Mdrop, Ndrop, pssDrop, ...
    betaFA, origin, beginAt)
% Function dropped = whenDropped(cost, Pd, Mdrop, Ndrop, pssDrop, ...
%                                betaNT, betaFA, origin, beginAt)
%       Apply the track drop logic to determine when (if) the track
%       reaches drop status.
% Input:
%   cost - [1xn] windowed cost vector (NOT the instantaneous cost)
%   Pd   - [1xn] probability of detection (assumed)
%   Mdrop - number of missed updates for the prototype benchmark track
%   Ndrop - window-of-regard length for the prototype benchmark track
%   pssDrop - steady state kinematic p(z|x)
%   betaFA - [1xn] assumed false alarm density
%   origin - Index of the first frame in which the track existed
%   beginAt - Index of first frame to consider a drop. Useful when dropping
%             shouldn't be considered until after confirmation.
%
% Output:
%   dropped - index of the frame in [1,n], or 0 if never dropped
%
% Andrew C. Rice, andrewcrice@gmail.com

% Ignore the first beginAt-1 frames
thresh(1:beginAt-1) = Inf;
% Compute a threshold for all subsequent frames
for winEnd = beginAt:length(cost)
    winStart = max([1,winEnd-Ndrop+1,origin]);
    if(winEnd-winStart+1) >= Ndrop
        %window is full
        Mnew = Mdrop;
        Nnew = Ndrop;
    else
        %window not yet full
        Nnew = winEnd-winStart+1;
        Mnew = Mdrop * (Nnew/Ndrop);
    end
    minPd = min(Pd(winStart:winEnd));
    thresh(winEnd) = Mnew * -log(1-minPd) + (Nnew-Mnew) * ...
        (-log(minPd/betaFA(winEnd))+pssDrop);
end

dropped = find(cost > thresh,1,'first');
```

```
43 if isempty(dropped)
       dropped = 0;
   end
```

Bibliography

1. "DoD Briefing on 2002 Advanced Concept Technology Demonstrations", 2002. URL http://www.defense.gov/Transcripts/Transcript.aspx?TranscriptID=2880. [accessed 9 May 2011].

2. *Reference Solar Spectral Irradiance: Air Mass 1.5*. Technical report, 2003. URL http://rredc.nrel.gov/solar/spectra/am1.5/. [accessed 19 May 2011].

3. "Exhibit R-2a, PB 2010 Defense Advanced Research Projects Agency, PE 0603767E: SENSOR TECHNOLOGY". *Research, Development, Test & Evaluation Project Justification*, 2009. URL http://www.dtic.mil/descriptivesum/Y2010/DARPA/0603767E.pdf. [accessed 21 May 2011].

4. *Nonconventional Exploitation Factors Data System (NEFDS)*. Technical report, 2009.

5. "Exhibit R-2, RDT&E Budget Item Justification: PB 2011 Air Force, PE 0305206F: Airborne Reconnaissance Systems". *Research & Development Descriptive Summaries*, 2010. URL http://www.dtic.mil/descriptivesum/Y2011/AirForce/0305206F_PB_2011.pdf. [accessed 25 January 2011].

6. "Exhibit R-2A, RDT&E Project Justification: PB 2012 Defense Advanced Research Projects Agency, PE 0603767E: SENSOR TECHNOLOGY". *Research & Development Descriptive Summaries*, 2011. URL http://www.js.pentagon.mil/descriptivesum/Y2012/DARPA/0603767E_3_PB_2012.pdf. [accessed 10 May 2011].

7. "U.S. Air Force Fact Sheet E-8C JOINT STARS". February 2011. URL http://www.af.mil/information/factsheets/factsheet.asp?id=100. [accessed 10 May 2011].

8. AFRL/XP. "Air Force Research Laboratory Business Plan FY08". 2008.

9. Baldridge, A.M., S.J. Hook, C.I. Grove, and G. Rivera. "The ASTER Spectral Library Version 2.0". *Remote Sensing of Environment*, 113(4):711–715, 2009.

10. Bar-Shalom, Y. and E. Tse. "Tracking in a Cluttered Environment with Probabilistic Data Association". *Automatica*, 11(5):451–460, 1975. ISSN 0005-1098.

11. Bischoff, S., M.J. Mendenhall, A.C. Rice, and J.R. Vasquez. "Adapting Learning Parameter Transition in the Generalized Learning Vector Quantization Family of Classifiers". *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on*, 1–4. IEEE.

12. Blackburn, J., M. Mendenhall, A. Rice, P. Shelnutt, N. Soliman, and J. Vasquez. "Feature Aided Tracking With Hyperspectral Imagery". *Proceedings of the SPIE, Conference on Signal and Data Processing of Small Targets*, 6699, August 2007.

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

14. Brakatsoulas, S., D. Pfoser, R. Salas, and C. Wenk. "On Map-Matching Vehicle Tracking Data". *Proceedings of the 31st International Conference on Very Large Data Bases*, 853–864. VLDB Endowment, 2005.

15. Civilis, A., C.S. Jensen, and S. Pakalnis. "Techniques for Efficient Road-Network-Based Tracking of Moving Objects". *IEEE Transactions on Knowledge and Data Engineering*, 698–712, 2005.

16. Clark, R.N. *USGS Digital Spectral Library splib06a*. US Geological Survey (USGS), 2007.

17. DeSieno, D. "Adding a Conscience to Competitive Learning". *Proceedings of the IEEE International Conference on Neural Networks I*, 117–124. New York, Jul 1988.

18. Duda, R.O., P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley & Sons, second edition, 2001.

19. Ellis, JM, HH Davis, and JA Zamudio. "Exploring for Onshore Oil Seeps with Hyperspectral Imaging". *Oil and Gas Journal*, 99(37):49–58, 2001. ISSN 0030-1388.

20. Fortmann, T., Y. Bar-Shalom, and M. Scheffe. "Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association". *Oceanic Engineering, IEEE Journal of*, 8(3):173–184, 1983. ISSN 0364-9059.

21. Green, R.O., M.L. Eastwood, C.M. Sarture, T.G. Chrien, M. Aronsson, B.J. Chippendale, J.A. Faust, B.E. Pavri, C.J. Chovit, M. Solis, M.R. Olah, and O. Williams. "Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)". *Remote Sensing of Environment*, 65(3):227–248, 1998. ISSN 0034-4257.

22. Grewal, M.S. and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley Online Library, 2001. ISBN 0471392545.

23. Guarino, D., B. Walls, and E. Miles. *Confirmatory Identification of Targets in Video*. Technical report, 2005.

24. Guo, Y., S. Hsu, H. Sawhney, R. Kumar, and C. Rao. *Video Object Fingerprinting for Confirmatory Identification (CID)*. Technical report, 2005.

25. Haboudane, D., J.R. Miller, E. Pattey, P.J. Zarco-Tejada, and I.B. Strachan. "Hyperspectral Vegetation Indices and Novel Algorithms for Predicting Green LAI of Crop Canopies: Modeling and Validation in the Context of Precision

Agriculture". *Remote Sensing of Environment*, 90(3):337–352, 2004. ISSN 0034-4257.

26. Hammer, B. and T. Villmann. "Generalized Relevance Learning Vector Quantization". *Neural Networks*, 15:1059–1068, 2002.

27. Hastie, T., R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer Verlag, 2009.

28. Howard, T.E., M.J. Mendenhall, and G.L. Peterson. "Abstracting GIS Layers From Hyperspectral Imagery". *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS'09. First Workshop on*, 1–4. IEEE, 2009.

29. Kalman, R.E. "A New Approach to Linear Filtering and Prediction Problems". *Journal of Basic Engineering*, 82(1):35–45, 1960.

30. Kanchanavally, S., R. Ordónez, and J. Layne. "Mobile Target Tracking by Networked Uninhabited Autonomous Vehicles Via Hospitability Maps". *American Control Conference, 2004. Proceedings of the 2004*, volume 6, 5570–5575. IEEE, 2004.

31. Kerekes, J.P., M.D. Presnar, K.D. Fourspring, Z. Ninkov, D.R. Pogorzala, A.D. Raisanen, A.C. Rice, J.R. Vasquez, J.P. Patel, R.T. MacIntyre, and S.D. Brown. "Sensor Modeling and Demonstration of a Multi-Object Spectrometer for Performance-Driven Sensing". *Proc. of SPIE Vol*, volume 7334, 73340J–1.

32. Kohonen, T. *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg, second edition, 2001.

33. Krajzewicz, D., M. Bonert, and P. Wagner. "The Open Source Traffic Simulation Package SUMO". *RoboCup 2006 Infrastructure Simulation Competition*, 2006.

34. Leininger, B. "A Next-Generation System Enables Persistent Surveillance of Wide Areas". *SPIE Newsroom*, April 2008. URL `http://spie.org/x23645.xml?ArticleID=x23645`. [accessed 10 May 2011].

35. Mendenhall, M.J. and E. Merényi. "Relevance-Based Feature Extraction from Hyperspectral Images". *IEEE Transactions on Neural Networks*, 19, Apr 2008.

36. Meyer, R.D., K.J. Kearney, Z. Ninkov, C.T. Cotton, P. Hammond, and B.D. Statt. "RITMOS: a Micromirror-Based Multi-Object Spectrometer". *Ground-based Instrumentation for Astronomy*, 5492(1):200–219, 2004.

37. Muster, R. *Exploitation of Geographic Information Systems for Vehicular Destination Prediction*. Master's thesis, Air Force Institute of Technology, 2009.

38. Neff, T. "Tall Order". *C4ISR Journal*, April 2010. URL `http://www.c4isrjournal.com/story.php?F=4452148`. [accessed 25 January 2011].

39. Paris, N.D. *LQG/LTR Tilt and Tip Control for the Starfire Optical Range 3.5 Meter Telescopes Adaptive Optics System*. Master's thesis, Air Force Institute of Technology, 2006.

40. Pendall, D.W. "Persistent Surveillance and Its Implications for the Common Operating Picture". *Military Review*, 85(6):41, 2005. ISSN 0026-4148.

41. Pierce, S. *Context Aided Tracking and Track Prediction in Aerial Video Surveillance*. Master's thesis, Air Force Institute of Technology, 2008.

42. Pyo, J.S., D.H. Shin, and T.K. Sung. "Development of a Map Matching Method Using the Multiple Hypothesis Technique". *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, 23–27. IEEE, 2001.

43. Reid, D. "An Algorithm for Tracking Multiple Targets". *Automatic Control, IEEE Transactions on*, 24(6):843–854, 1979. ISSN 0018-9286.

44. Ren, H. and C.I. Chang. "Automatic Spectral Target Recognition in Hyperspectral Imagery". *Aerospace and Electronic Systems, IEEE Transactions on*, 39(4):1232–1249, 2003.

45. Rice, A.C., J.R. Vasquez, J.P. Kerekes, and M.J. Mendenhall. "Persistent Hyperspectral Adaptive Multi-Modal Feature-Aided Tracking". *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, 7334(1):73340M, 2009.

46. Robertson, D.C., A. Berk, and L.S. Bernstein. "MODTRAN: A Moderate Resolution Model for LOWTRAN 7", 1989.

47. Romberg, J. "Imaging via Compressive Sampling". *Signal Processing Magazine, IEEE*, 25(2):14–20, 2008.

48. Rose, L.J. "Air Force Research Laboratory's Focused Long Term Challenges". volume 6981. SPIE, 2008.

49. Salem, F. and M. Kafatos. "Hyperspectral Image Analysis for Oil Spill Mitigation". *Paper presented at the 22nd Asian Conference on Remote Sensing*, volume 5, 9. 2001.

50. Sato, A. and K. Yamada. "Generalized Learning Vector Quantization". David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo (editors), *Advances in Neural Information Processing Systems 8: Proceedings of the 1995 Conference*, 423–429. MIT Press, Cambridge, MA, 1996.

51. Schott, J.R. *Remote Sensing: the Image Chain Approach, 2nd ed.* Oxford University Press, Oxford, NY, 2007.

52. Schott, J.R., S.D. Brown, R.V. Raqueno, H.N. Gross, and G. Robinson. "Advanced Synthetic Image Generation Models and Their Application to Multi-/Hyperspectral Algorithm Development". *27th AIPR Workshop: Advances in Computer-Assisted Recognition*, 3584(1):211–220, 1999.

53. Secrest, B.R. and J.R. Vasquez. "Optimal Spatial Sampling of Hyperspectral Imagery for Fusion with Panchromatic Video in Multitarget Tracking". *Sensors Applications Symposium, 2009. SAS 2009. IEEE*, 255–260. IEEE, 2009.

54. Soliman, N. *Hyperspectral-Augmented Target Tracking*. Master's thesis, Air Force Institute of Technology, 2008.

55. Tahk, M. and J.L. Speyer. "Target Tracking Problems Subject to Kinematic Constraints". *Automatic Control, IEEE Transactions on*, 35(3):324–326, 1990.

56. Thenkabail, P.S., R.B. Smith, and E. DePauw. "Hyperspectral Vegetation Indices and Their Relationships with Agricultural Crop Characteristics". *Remote Sens. Environ.*, 71, 2000.

57. Wald, A. and J. Wolfowitz. "Optimum Character of the Sequential Probability Ratio Test". *The Annals of Mathematical Statistics*, 19(3):326–339, 1948. ISSN 0003-4851.

58. Yuhas, R.H., A.F.H. Goetz, and J.W. Boardman. "Discrimination Among Semi-Arid Landscape Endmembers Using the Spectral Angle Mapper (SAM) Algorithm". *Summaries of the Third Annual JPL Airborne Geoscience Workshop*, volume 1, 147–149. Pasadena, CA: JPL Publication, 1992.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
16–06–2011	Master's Thesis	June 2007 — June 2011

4. TITLE AND SUBTITLE

Context-Aided Tracking with
Adaptive Hyperspectral Imagery

5a. CONTRACT NUMBER

DACA99–99–C–9999

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Andrew C. Rice

5d. PROJECT NUMBER

n/a

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/11-43

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

Air Force Research Laboratory (Karmon M. Vongsy)
2241 Avionics Circle
Wright-Patterson Air Force Base, OH 45433
(937)528-8285 Karmon.Vongsy@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RYAT

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

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

A methodology for the context-aided tracking of ground vehicles in remote airborne imagery is developed in which a background model is inferred from hyperspectral imagery. The materials comprising the background of a scene are remotely identified and lead to this model. Two model formation processes are developed: a manual method, and method that exploits an emerging adaptive, multiple-object-spectrometer instrument. A semi-automated background modeling approach is shown to arrive at a reasonable background model with minimal operator intervention. A novel, adaptive, and autonomous approach uses a new type of adaptive hyperspectral sensor, and converges to a 66% correct background model in 5% the time of the baseline – a 95% reduction in sensor acquisition time. A multiple-hypothesis-tracker is incorporated, which utilizes background statistics to form track costs and associated track maintenance thresholds. The context-aided system is demonstrated in a high-fidelity tracking testbed, and reduces track identity error by 30%.

15. SUBJECT TERMS

tracking, context-aided tracking, hyperspectral, hyperspectral imagery, hyperspectral exploitation, adaptive hyperspectral imagery, remote sensing

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Juan R. Vasquez
U	U	U	UU	156	19b. TELEPHONE NUMBER *(include area code)* (937)286-7580; Juan.Vasquez@wpafb.af.mil