3-1-2008

Multispectral persistent surveillance

Andrew J. Adams

Multispectral Persistent Surveillance

by

Andrew J. Adams

B.S. University of Colorado, 1991
M.S. University of Colorado, 1992

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
Rochester Institute of Technology

March 11, 2008

Signature of the Author _____

Accepted by _____
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

The Ph.D. Degree Dissertation of Andrew J. Adams
has been examined and approved by the
dissertation committee as satisfactory for the
dissertation required for the
Ph.D. degree in Imaging Science

Dr. John R. Schott, Dissertation Advisor

Dr. Harvey E. Rhody

Dr. David W. Messinger

Dr. David S. Ross

Date

DISSERTATION RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE

Title of Dissertation:

Multispectral Persistent Surveillance

I, Andrew J. Adams, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use or profit.

Signature _____

Date

Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: **Multispectral Persistent Surveillance**

Name of author: Andrew John Adams
Degree: Doctor of Philosophy
Program: Imaging Science
College: Science

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, Andrew John Adams, hereby **grant permission** to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: _____ Date: 04/07/2007

Print Reproduction Permission Denied:

I, _____, hereby **deny permission** to the RIT Library of the Rochester Institute of Technology to reproduce my print thesis or dissertation in whole or in part.

Signature of Author: _____ Date: _____

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, Andrew John Adams, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: _____ Date: 04/07/2007

DISCLAIMER

The views expressed in this dissertation 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.

Acknowledgments

I would like to thank my committee: Dr. John Schott, Dr. Dave Messinger, Dr. Harvey Rhody, and Dr. David Ross. I especially appreciate the drive and vision provided by Dr. Schott and technical guidance from Dr. Messinger. The original algorithms were provided by Dr. Roland Miezianko, without whose help this project could not have been started. Thank you to all of the other USAF AFIT officers and CIS students for helping me see the obvious. A special thank you to Lt Col Marcus Stefanou, whose mentorship and guidance taught me to have faith in myself and others. Finally, to my wife and two daughters: Alison, Annabelle, and Imogen – without your love and support none of this would have been possible.

Multispectral Persistent Surveillance

by

Andrew J. Adams

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

The goal of a successful surveillance system to achieve persistence is to track everything that moves, all of the time, over the entire area of interest. The thrust of this thesis is to identify and improve upon the motion detection and object association aspect of this challenge by adding spectral information to the equation. Traditional motion detection and tracking systems rely primarily on single-band grayscale video, while more current research has focused on sensor fusion, specifically combining visible and IR data sources. A further challenge in covering an entire area of responsibility (AOR) is a limited sensor field of view, which can be overcome by either adding more sensors or multi-tasking a single sensor over multiple areas at a reduced frame rate. As an essential tool for sensor design and mission development, a trade study was conducted to measure the potential advantages of adding spectral bands of information in a single sensor with the intention of reducing sensor frame rates. Thus, traditional motion detection and object association algorithms were modified to evaluate system performance using five spectral bands (visible through thermal IR), while adjusting frame rate as a second variable. The goal of this research was to produce an evaluation of system performance as a function of the number of bands and frame rate. As such, performance surfaces were generated to assess relative performance as a function of the number of bands and frame rate.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The challenge of persistent surveillance has become increasingly important in terms of national security in light of world events over the last few years. The notion of persistent surveillance is wide reaching and multifaceted with several different interpretations. The Department of Defense defines the term as "…the ability of collection systems to linger on demand in an area to detect, locate, characterize, identify, track, [and] target…to deter or forestall anticipated adversary courses of action" [DOD:2006] Another, possibly more useful definition of persistent surveillance emphasizes "sensing suites that tailor their observations to the adversary's rate of activity." [Signal:2002] The goal for any successful surveillance system to achieve persistence is to track everything that moves, all of the time, over the entire area of interest.

The thrust of this thesis is to identify and improve upon the motion detection and object association aspect of this challenge by adding spectral information to the equation. In order to tackle this problem, it is helpful to break it into a workable subset of challenges. First of all, consider a large geographic area of responsibility (AOR) for which a commander has the task of monitoring all activity, as depicted in figure 1.1.

Now consider that the AOR is too large to cover with a single sensor—say a video camera running at a typical 30 frames per second (fps)—due to the field of view of the camera. There are several approaches to achieve persistent surveillance, one of which would be to simply employ more sensors of the same type and divide the area into pieces. Another approach would be to use a rotating or scanning mirror system at the front end of a single sensor and spend less time (or fps) monitoring each sub-area, as shown in figure 1.2.

FIGURE 1.1 –Area of Responsibility (AOR).

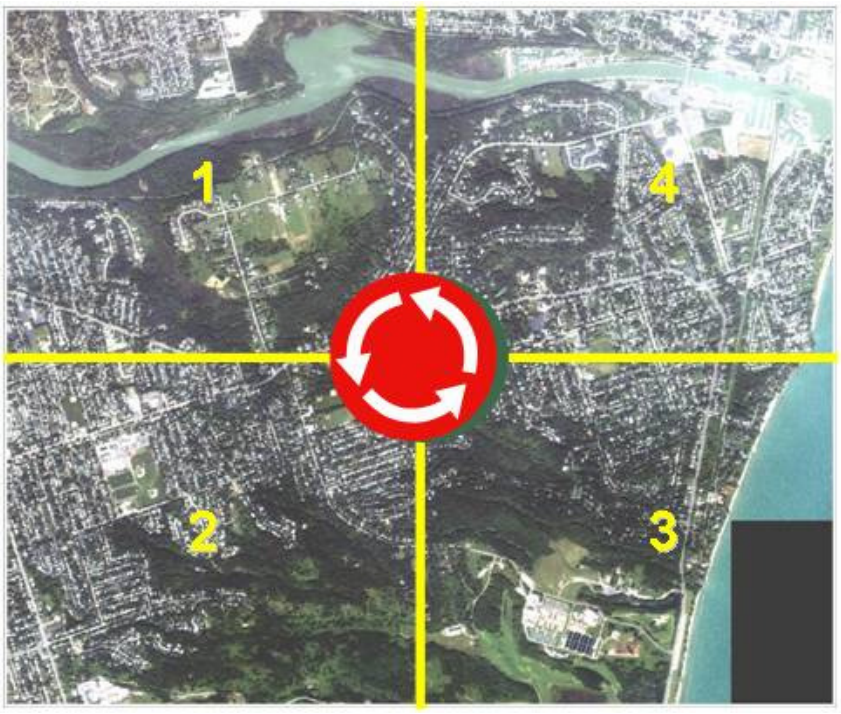


FIGURE 1.2 – AOR Divided into Four Regions (i.e. Spinning Mirror).

The challenge then becomes one of achieving the same detection and tracking performance at a reduced frame rate (in this example, about 7 fps over four separate areas). Based on the simple premise that adding spectral information to the data collection should enhance detection and tracking performance, a multispectral sensor might be able to achieve the required performance at a reduced frame rate. In order to investigate this premise, we need to take a closer look at the detection and tracking tasks.

The system model of a typical surveillance system, as depicted in figure 1.3, shows three top-level tasks as moving object detection, object segmentation, and object association. In order to detect a moving object, some means of determining changes between data frames is needed. Regardless of the technique, non-moving (background) pixels are distinguished from moving (foreground) pixels. Groups of moving pixels can then be segmented into objects based on their common characteristics, such as appearance, velocity, and location. Finally, in order to establish a track, each segmented object needs to be associated to an object in the next frame.

Once a moving object has been associated from one frame to the next, second-level tasks can be performed. Depending on the objectives of the system, tasks such as track management, object classification, and behavior analysis are accomplished. Finally, these analyses can be followed by a third-level automated output or response, such as calling in further sensor assets or adjusting the current sensor allocation to a sub-region of heightened interest. The focus of this thesis is in the top-level tasks of moving object detection, segmentation, and association (as identified by the green components in the system diagram in figure 1.3).

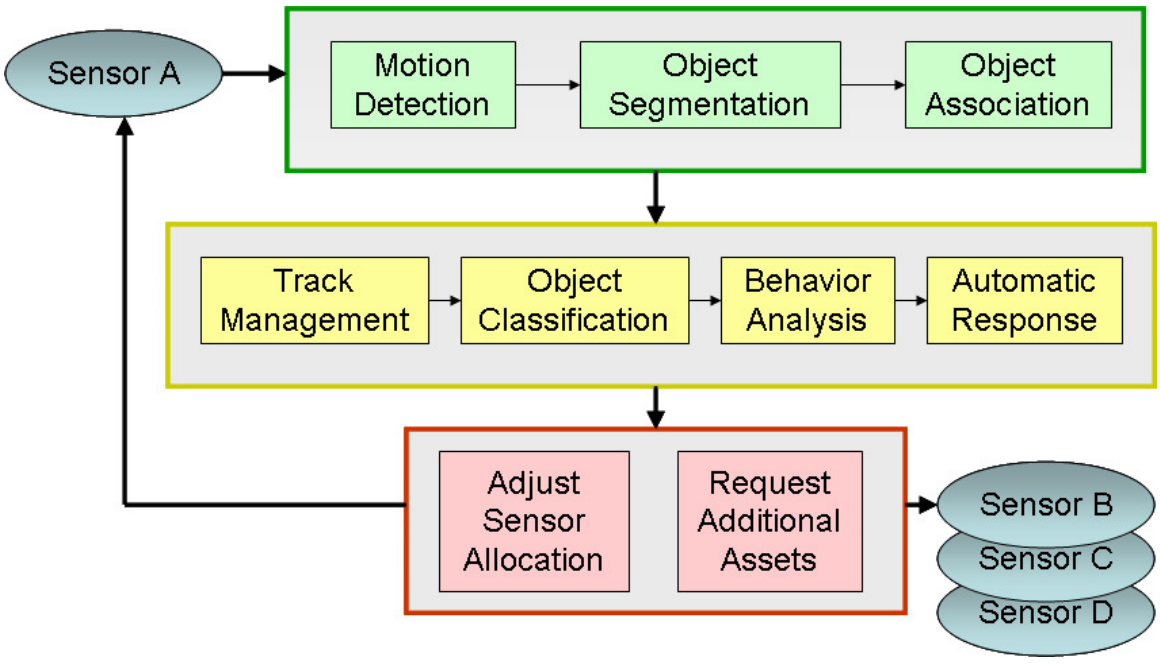


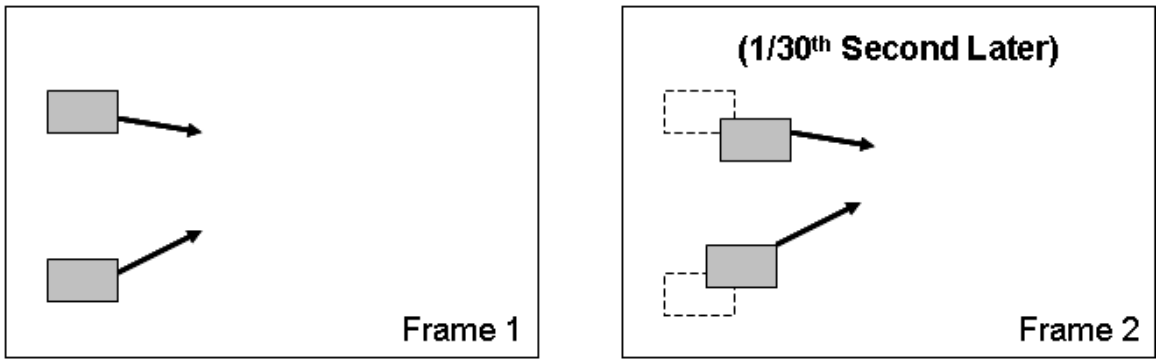
FIGURE 1.3 – Surveillance System Model.

The premise of adding spectral information to current single-band techniques has merit, making improved performance at reduced frame rates seem plausible. In each of the three tasks identified (detection, segmentation, and association), additional spectral content provides useful information. First, in the case of moving object (or change) detection, typical single-band systems rely on a single grayscale value per pixel to decide if something has changed from one frame to the next. By adding additional bands of information to each pixel, change detection becomes more sensitive to subtle changes and more discriminating to non-important changes.

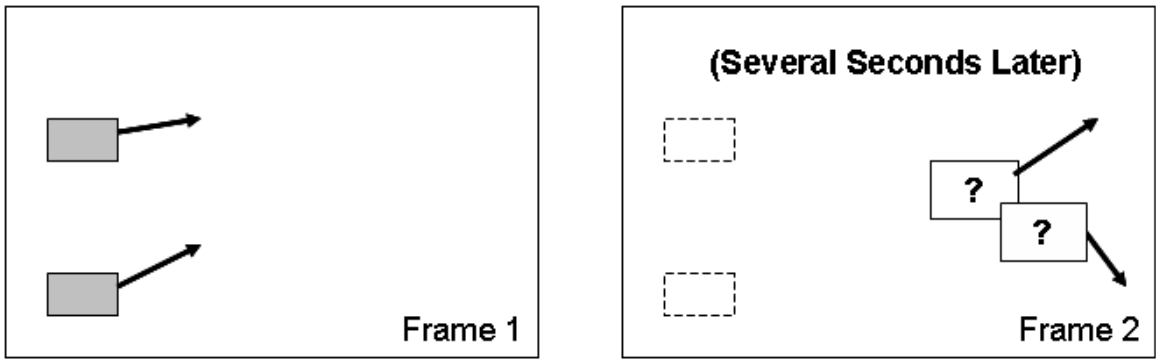
The simplest example is an area of pixels in one frame which may have the same grayscale appearance as in the next frame, but in reality a person wearing a red sweater is crossing in front of a red-brick building. Such a change might be missed using a single-band (or even color) sensor. However, given an additional thermal band, the bright (hot) person would stand out from the dark (cold) building.

Once moving pixels are identified, object segmentation in a single-band system relies primarily on pixel location, or proximity, to group pixels into an object. Additional information such as common velocity or appearance might help to distinguish if two objects have come together, or even if one object is temporarily obscuring the other. However, at reduced frame rates, the position and velocity estimates become unreliable. Again, by adding spectral detail to the appearance model of each object, distinguishing two or more objects becomes easier. Finally, object association in a single-band system typically becomes a task of spatial comparison of brightness value distributions and object characteristics such as velocity. Even the simplest spectral techniques such as spectral angle mapping (SAM) could provide a significant advantage over single-band techniques in object discrimination. In this case, instead of modifying existing single-band techniques to include spectral data, we can apply an additional filtering step such that we can compare objects spectrally first and then spatially.

To illustrate the idea of adding spectral information to a low frame rate collection, consider the simplistic example in figure 1.4, where a single-band sensor fails to associate objects at a low frame rate due to an ambiguity between objects in the next frame. Using typical video rate data at 30 fps, as seen in figure 1.4a, moving objects tend to be very close to the previous location making object association easier. However, at a reduced frame rate—say one frame every few seconds—the objects are more sparsely located and cannot be uniquely identified, as seen in figure 1.4b. Because a significant amount of time has passed between frames, the two moving objects are now overlapping and cannot be distinguished from one another. Furthermore, changes since the last observation may have occurred such that prior trajectories and grayscale appearance are not sufficient to resolve the objects.



(a) Single-band at 30fps



(b)Single-band at low fps

FIGURE 1.4– Moving object ambiguity (grayscale).

However, by adding spectral detail to each object (color in this simple example), as seen in figure 1.5, the ambiguity is resolved and the two objects can be distinguished.

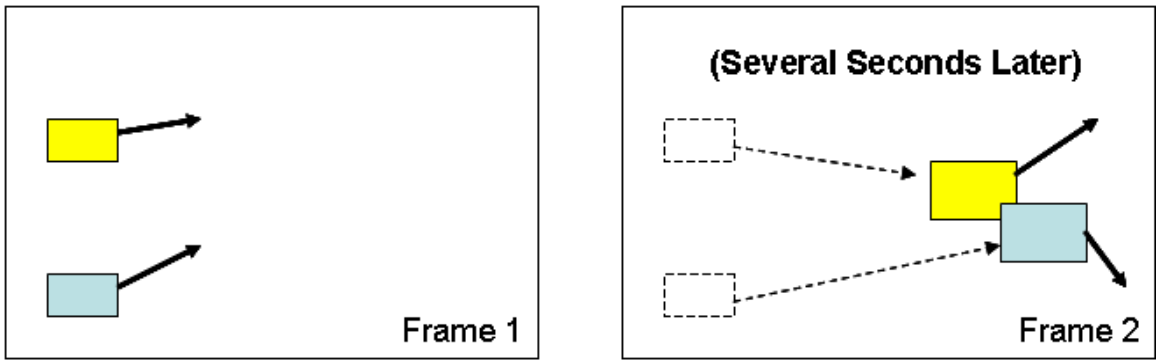


FIGURE 1.5 – Moving objects not ambiguous at low fps (color).

Current research has not emphasized this particular aspect of persistent surveillance, presenting an opportunity for novel work. An exhaustive review of current motion detection and tracking research shows two related areas of interest: multi-sensor fusion (particularly in visible/IR) and low frame rate, visible-band tracking. However, there doesn't appear to be any current research in the combination of the two ideas: Using multispectral data to enable low frame rate methods.

Based on the above discussion, the objective of this research was to evaluate the impact of including spectral information in current motion detection and object association algorithms with specific emphasis on reduced frame rates. Two different video datasets were used: one synthetic and the other real world data. The synthetic dataset was developed for this project by using Digital Imaging and Remote Sensing Image Generation (DIRSIG). The real world dataset was collected using the WASPLITE system; a portable version of the Wildfire Airborne Sensor Program (WASP) sensor platform. Measures of performance as a function of spectral and temporal resolution were also developed. The trade study was successfully assembled to evaluate the potential advantages to be gained by replacing current single-band video surveillance systems with multispectral sensors.

Thus, the stated hypothesis of this research is simply this: By adding more spectral information to the data, better system performance can be achieved at low frame rates than with a single-band system. It is important to note here that an operational assumption has been made from the outset of this study: In the early stages of a persistent surveillance system, missed objects are worse than false alarms. In other words, we want to catch everything that moves at the expense of tolerating more false alarms.

Chapter 2

Background

A review of existing methods for moving object detection and tracking provides a framework for determining the methodology for this research. Algorithm selection was based on three criteria: First, to find the most recent, "state of the art" techniques; Second, to determine which techniques would adapt well to multispectral data; Third, to allow for low frame rate input data. The background section is organized into three general areas of interest: moving object detection; object association and tracking; and current research in visible/IR fusion and low frame rate tracking methods.

2.1 Moving Object Detection

When considering the detection of moving objects in a scene an important distinction must be made based on the time interval between observations. The earliest work in processing multiple collections of the same scene occurred prior to the appearance of video surveillance. The distinction between change detection and motion detection stems from their different objectives. Whereas change detection techniques attempt to determine large scale changes in a scene over large time intervals, motion detection methods operate on a very small time scale and attempt to estimate the position and velocity of moving objects.

2.1.1 Change Detection

The detection of moving objects in video surveillance evolved from basic change detection methods developed for comparing two or more images widely spaced in time. For instance, much of the early work was used to compare Landsat images to determine seasonal or even annual changes in a certain region. These so called "multitemporal techniques" focused

primarily on image differencing and required the images to be co-registered. These techniques were based on statistical measures of similarity between images [Kawamura:1971] and segmentation by region matching using features such as size, shape, spatial and even spectral properties [Price:1977]. Other multi-temporal techniques used temporal trend analysis [Engvall :1977] and Principal Component Analysis [Byrne:1980], both of which were applied to Landsat data. As change detection techniques evolved, more sophisticated approaches to image differencing techniques emerged. Variations on determining an appropriate change threshold led to measuring the change in other image quantities such as the entropy of the histogram [Kapur:1985] and intensity gradients [Parker:1991]. Difficulties in detecting change in remotely-sensed images can arise from misregistration [Townsend:1992] [Bruzzone:1997], or drastic changes in lighting, atmosphere, and sensor calibration between the two acquisition dates [Singh:1989].

More recent work has continued to improve unsupervised change detection by applying a simple yet adaptive decision threshold [Bruzzone:2002], using the assumption that the histogram of the difference image can be modeled as a mixture density of two classes: changed and unchanged pixels. The difference image (*XD*) is defined as the magnitude of the spectral change vector, computed for pixel *(i, j)* as shown in equation 2.1, where *X1* and *X2* are vectors of brightness values at selected bands,

$$XD(i, j) = \| X1(i, j) - X2(i, j) \| \ . \tag{2.1}$$

This technique was applied to data collected by a passive multispectral scanner installed on a satellite (Wide Field Sensor (WiFS) on the IRS-P3 satellite, where *X1* and *X2* are multi-temporal samples of the same location on two captured images. A large value for *XD* indicates a changed pixel, whereas a small value is an unchanged pixel. In this way, *XD* is modeled as a mixture density of changed or unchanged pixels. The method is adaptive in that it does not assume an a priori model of the data distribution and semiparametric because Bayesian decision theory is used to determine the correct mixture.

Of particular interest in this example, the process uses change vector analysis (CVA) to generate the difference image. In this case, each pixel in the image is represented by a spectral vector and each pair or corresponding pixels in the two images produces a "spectral change vector". Using the magnitude of the change vector at each pixel, the resulting spectral change map produces a grayscale image where higher values indicate greater change. Spectral change mapping will be considered further in the methodology section of this thesis.

Another area of research regarding change detection is in support of wide area surveillance where detection of new activities and events are monitored over very large geographic areas. Here again, the intention is not to monitor moving objects in real-time, but to determine large scale changes over long periods of time. In recent work applied to Landsat images [Carlotto:1997], an attempt was made to overcome the problems associated with changes in solar angle, sensor gain, atmospheric scattering, path radiance and other environmental conditions.

The technique does not derive patterns of change directly from the observed brightness values. Instead, adaptive techniques use information over larger areas in a sliding window to model and predict one image from another, using what is called forward/backward prediction. In this way, subsequent observations of the same location can be put in the same frame of reference as the original observation. Thus, the difference between the original and predicted images is used as a measure of change. However, techniques such as this are attempting to resolve the change in conditions between acquisitions when the time interval is significantly greater than video surveillance frame rates. Motion detection methods applied to video are an evolution from change detection, whereby at video frame rates the scene and sensor conditions have not changed significantly between frames.

2.1.2 Motion Detection

A distinction can be made between change detection and motion detection based on the period of time between images. In modern video surveillance the standard frame rate of 30 frames per second (fps) provides a very accurate model of the scene over multiple frames. In this case we have the benefit of little environmental change between observations. Additionally, because of the short time between observations, the majority of change between frames can be interpreted as motion. Similar to change detection, the discriminating factor is in detecting moving and non-moving pixels, which can be considered as foreground (or target) and background, respectively. Similar to the distinction made in the previous section [Bruzzone:2002], the problem becomes a two-class system: moving and non-moving pixels.

2.1.2.1 Fundamental Techniques

As reviewed in the DARPA Video Surveillance and Monitoring (VSAM) report [Collins:2000], there are essentially three basic approaches to motion detection: temporal differencing [Anderson:1985]; background subtraction [Haritaoglu:1998] [Wren:1997]; and optical flow [Lucas_Kanade:1981] [Barron:1994]. Although temporal differencing is straightforward and is adaptive to dynamic environments, it doesn't always extract all relevant feature pixels. Conversely, background subtraction generally provides complete feature data but is adversely sensitive to dynamic scene changes such as variation in lighting. Furthermore, background subtraction requires training observations in order to build up a background model. Optical flow is essentially a motion estimation technique useful in that it can detect moving objects in the presence of camera motion; however, it is computationally expensive, assumes constant velocity, and requires a suitable threshold to discriminate moving objects. Of the three fundamental motion detection methods, background subtraction appears to be the most widely used due to its simplicity and robustness [Porikli:2005].

2.1.2.2 Motion Salience

Regardless of the method of determining which pixels are moving, there is a further distinction to be made: Which of the detected moving pixels do we care about? False alarms can result

from "motion clutter" [Collins:2000] resulting from distractions such as objects blowing in the wind, moving shadows, or sensor noise. Salience of moving objects can be used for filtering out distracting, unimportant motion. Salient motion can be defined by objects with directionally consistent motion such that only objects moving "with a purpose" are detected [Wixson:2000]. Therefore, a surveillance system should also rely on motion salience for false alarm rejection. Although this technique was not applied in the methodology for this trade study, it is a point of interest for future versions. As a graduate-course project, motion salience was applied to grayscale video data to determine the feasibility of implementing such a filter [Adams:2006].

A simplified method to determine if objects are directionally consistent applies optical flow to the difference images as opposed to the original frames [Tian:2005]. The algorithm employs a temporal filter on the pre-processed scene to determine the flow field properties over time (typically 10 frames). The filtered scene then highlights only the salient objects, ignoring the non-interesting motion.

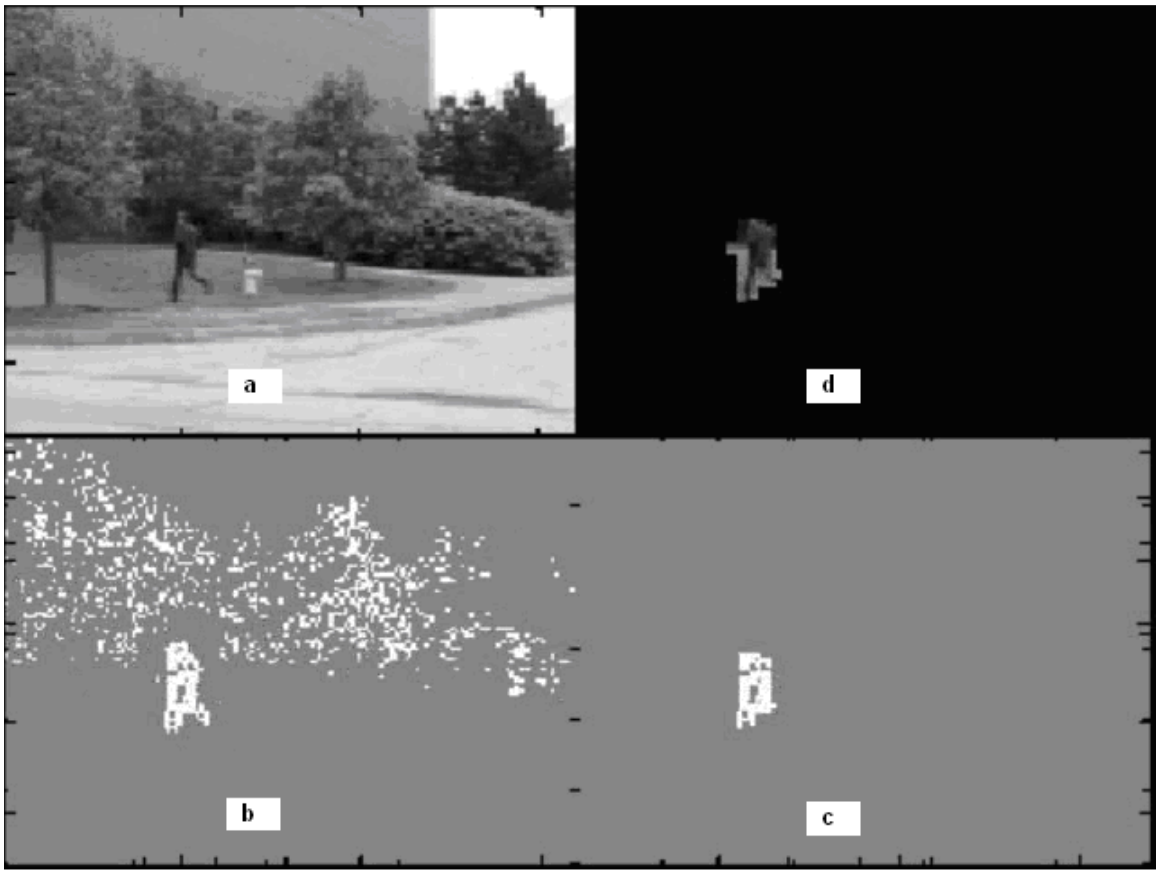


FIGURE 2.1 – Salient Motion Mask Removes Distracting Motion [Adams:2006].

The example shown in figure 2.1 demonstrates the process of using a salient motion mask (counter-clockwise starting in the upper left image). The upper left scene (a) shows one of the original frames on a windy day. The lower left (b) shows the difference image, depicting motion not only in the running figure but in the swaying trees. The lower right (c) shows the results of

the temporal filter which determines which pixels had a consistent motion over a period of 10 frames. Finally, the upper right corner (d) shows the masked scene with only the running figure isolated from the rest of the distracting motion.

2.1.2.3 Hybrid Techniques

The most recent strategies in motion detection apply a hybrid approach to the three fundamental techniques. The VSAM system [Collins:2000] employs frame differencing and adaptive background subtraction with some success, by first detecting motion using three-frame differencing then extracting region information based on an adaptive background model. Motion detection is then accomplished in a layered approach by first conducting a pixel analysis to determine moving pixels, followed by a region analysis to decide if a detected object is still moving or temporarily stationary. Another very interesting hybrid approach employs a threshold to spatiotemporal entropy [Jing:2004], where each pixel is described by the entropy of an accumulated histogram of brightness values in a local window (spatial) over several frames (temporal). However, spatial structure (i.e. edge pixels) affects the histogram adversely. The hybrid solution was to accumulate the histogram from the difference image between consecutive frames. Although the detection results were promising, the approach was overly complicated, computationally expensive, and still sensitive to illumination changes. Furthermore, this technique did not appear to be easily extended to multispectral data.

2.1.3 Spatiotemporal Texture Vectors

There is another such hybrid strategy which observes local variability in a spatiotemporal sense, also detecting motion where spatiotemporal variability exceeds a local threshold. In this other approach, the spatiotemporal description of each pixel is cast into a local "texture vector" and simplified using principal component analysis. Because of this inherent step to reduce dimensionality, spatiotemporal texture vectors appear to be an ideal choice for extending the technique to multispectral data.

The novel concept of using spatiotemporal texture vectors [Latecki_Miezianko:2006] to detect local variability in space and time seems promising. Of the three fundamental techniques listed in section [2.1.2.1], this method most closely resembles background subtraction in that it requires training observations (assuming no motion) to develop a model of background behavior. However, it has been shown to outperform the most popular background subtraction technique which uses a Gaussian mixture model to model the background [Stauffer_Grimson:1999].

In the spatiotemporal vector technique, a video sequence is reconstructed into a set of three dimensional blocks consisting of a local two dimensional window (spatial) over several frames (temporal), as illustrated in figure 2.2. Using the example of an 8 x 8 pixel window over 3 temporal frames, each texture vector consists of 192 brightness values. Thus, each spatiotemporal region of the scene is a local subset of the entire scene and is described by a 192-element spatiotemporal texture (SP) vector.

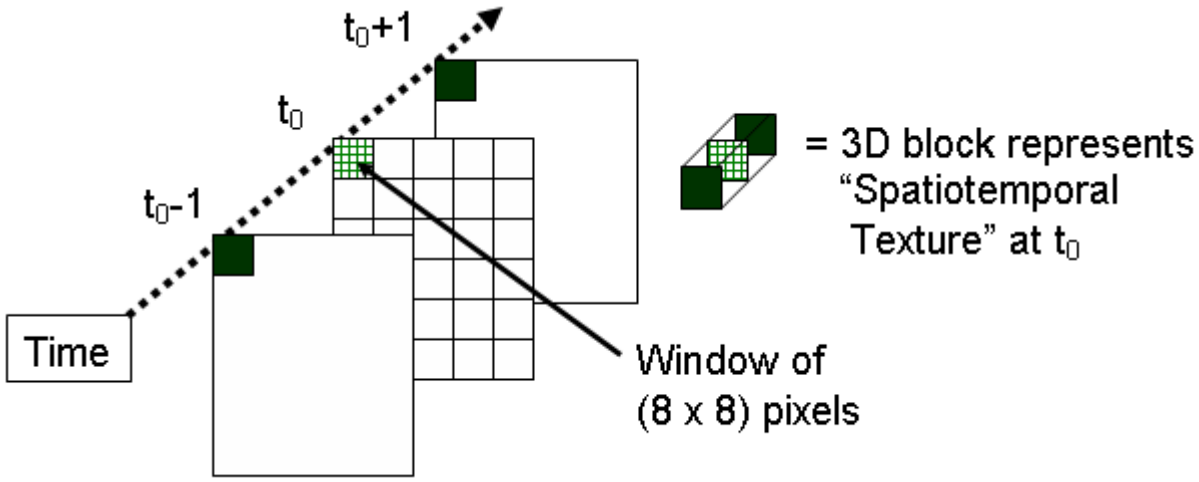


FIGURE 2.2 – Spatiotemporal Texture at a Single Block.

The dimensionality of each SP-vector can then be reduced by applying principal component analysis using a 192 x 192 covariance matrix. The covariance matrix is estimated based on multiple observations of the same two dimensional region of the scene over an initialization period. Consequently, each 8 x 8 spatial window at a given time (t_0 in figure 2.2) is represented by a 10-element vector by keeping only the first ten principal components.

Motion detection is then achieved by applying a dynamic threshold to local variation in spatiotemporal texture space, tagging all the pixels in the 8 x 8 window as moving if the local behavior is inconsistent with the background model. As with traditional background subtraction techniques, this approach relies on an initialization period assuming no motion in the scene to establish the background model. Once the detector is running on an active scene, the background model is updated only when a pixel is determined to be stationary.

Detecting motion based on spatiotemporal variability may be difficult to grasp intuitively, especially after the dimensional reduction of the texture vectors. The spatiotemporal texture at a given location is now described by an (N x 10) array where N is the total number of frames in the video sequence. To assist in visualizing how to measure motion at each location, Miezanko uses the notion of "motion orbits" by plotting the first three principal components as a function of time, as seen in figure 2.3 [Miezanko:2006]. The central cluster of blue dots in the figure represents time intervals when the location under observation is relatively stationary. However, when a moving object passes through that location, the principal component values increase due to local variation in texture. Thus, motion in the scene causes relatively rapid changes in the motion orbits as depicted by the red dots in the figure. All of the pixels in the 8x8 block at those time intervals would be labeled as moving.

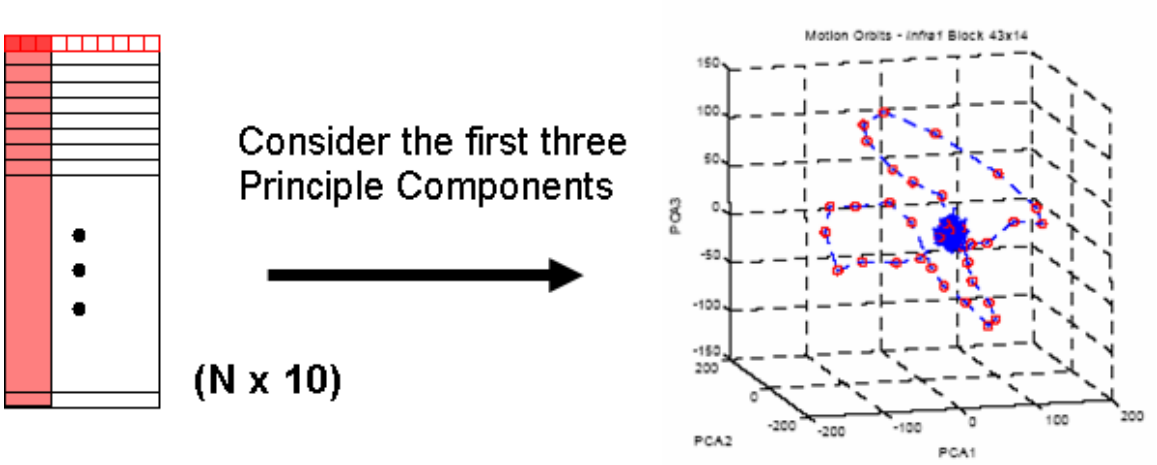


FIGURE 2.3 – Motion Orbits Demonstrate Spatiotemporal Variability [Miezanko:2006].

The most attractive aspect of this technique is the data structure is inherently able to include multiple spectral bands simply by extending the length of the SP-vectors. Given a motion detection technique that can be applied to multispectral data, the next task is segmenting the moving pixels into moving objects. Once all moving objects have been identified in each frame, the critical task in tracking them is to make an association between objects from one frame to the next—a task which should also be made easier given the advantage of multispectral data to discriminate between different objects.

2.2 Object Segmentation

In order to perform the next step of associating detected objects from one frame to the next, object segmentation is a necessary processing step. Once all pixels (or square regions) of a frame are labeled as moving or not moving, the moving pixels can be grouped into contiguous objects based on proximity, connectivity, and appearance. Although there is a tremendous amount of literature focused on this specific topic, it was not considered a primary area of research for this project. A more detailed description of the segmentation scheme is provided in the methodology section.

2.3 Object Association and Tracking

The ultimate goal of the detection and tracking system is to provide a stable track for each object. The basic principles of tracking fall under two areas: object matching and motion models. The two principles are intertwined to produce effective object tracks, but the main idea is to predict an object's location in the next frame using a motion model, then match objects based on the prediction and various object qualities. Although a stable track evolves from effective track management, the accuracy of a tracking system depends upon the confidence with which objects

are matched from one frame to the next [Hu:2004 – Survey]. Also called object association, there are four major categories of methods for matching objects: model-based, contour-based, feature-based, and region-based. Any combination of these four methods can be considered a fifth, catch-all means of matching objects, such as using both regional variations in the image and specific object features [Cavallaro:2005].

Once objects have been matched from one frame to the next, track management consists of track-labeling tasks such as track initiation, splitting, merging, updating, and termination. In this trade study, track management is considered a second level function (as depicted in the system diagram in figure 1.3). As such, an overview of track management approaches will be presented, with the intention of selecting a suitable method to track multiple moving objects. The emphasis, however, is on object association, as it stands to show the greatest improvement by adding spectral information to object descriptions. Furthermore, when considering low frame rate data, motion models tend to fail due to the dynamic nature of the moving objects. In this case, we rely even more heavily on the spectral appearance model of the objects to assist in object association rather than the uncertain predictions of location and velocity.

2.3.1 Object Association Methods

The most crucial step in tracking a moving object is deciding which track it belongs to—also called object association. As reviewed by [Cavallaro:2005] and [Hu:2004], there are four basic methods for performing the object association task. The first method, model-based matching, requires *a priori* knowledge of object shape. Although it handles partially occluded objects based on the fidelity of the model, it is computationally expensive and is limited to the database of objects [Koller:1993]. The second method, contour-based matching, tracks only region boundaries by using "snakes" or meshes that allow for deformable objects. However, because the technique requires a complete contour, it is unable to track partial occlusions [Peterfreund:1998][Gnsel:1998]. The third method, feature-based matching uses spatial features such as edges, line segments, or corners to uniquely match objects. Although tracking a portion or subset of an object allows for partial occlusions, grouping by features makes object identification difficult [Beymer:1997]. The fourth method, appearance-based matching, uses object characteristics such as color and texture. Appearance-based methods are similar to the feature-based methods because they both rely on neighboring pixels and fail to track complex deformations [Meier:1998][Tao:2002]. However, appearance-based matching uses spectral information within the region as well as spatial, which makes it the most attractive technique for processing multispectral data.

Probably the most straightforward (and thus most popular) technique of appearance-based matching can be referred to as block-matching [Tekalp:1995]. In this case, the displacement of a pixel in the previous frame is estimated by searching a window around that location in the current frame. The search is usually limited to a search window around the previous location due to computational limits. Block-matching algorithms can vary in matching criteria, search strategy, and block size. An extension of block-matching is provided in the DARPA Video Surveillance and Monitoring (VSAM) system [Collins:2000], where matching is performed with image correlation, computed by convolving the target's intensity template over candidate regions

in the next frame. Essentially, the displacement d of a target is estimated by accumulating a weighted sum of absolute intensity differences between a region in the previous (target) frame and a region in the current (candidate) frame. The best position match is given by the displacement $\hat{d}$ that minimizes the correlation. This technique will be discussed in greater detail in the methodology section.

A fifth category of object matching techniques can be derived as a hybrid of any of the four above techniques. One example of a hybrid presented by [Cavallaro:2005] proposed a hierarchical approach where the object features were first used for initial object segmentation. Next, region appearance values such as color, texture, and optical flow were used to describe the local area surrounding each object. In this case, each region was represented by region descriptors which were finally used for data association and track labeling. This example provides good insight to a future direction in hybrid object association and tracking methods—one which will emphasize local spatiotemporal variability combined with spectral matching. In fact, as described in the future work section, a proposed method combines these ideas into a three-step process (see appendix A). The first step compares the spectral similarity of objects, assigning a score. The second step converts the multispectral data into an optimized grayscale map based on distance to the spectral mean. The third step compares the grayscale objects in the VSAM single-band method mentioned above. As a result, the combined spectral and spatial similarity scores should provide a higher fidelity matching scheme than current techniques.

2.3.2 Spectral Matching

With the intention of using spectral information as the first step in object matching, we can expand the appearance-based methods to include spectral target detection techniques. Even the simplest techniques such as Spectral Angle Mapping (SAM) [Yuhas:1992] should produce additional information for matching not accounted for in the single-band methods. SAM compares two pixels by computing a spectral angle between each spectrum, as shown in equation 2.2, where x and y are two multidimensional spectra,

$$\text{SAM}\ (x, y) = \cos^{-1}\left[\frac{x \bullet y}{|x||y|}\right] \quad . \tag{2.2}$$

Consider the two-band example in figure 2.4, where a pixel spectrum is compared to a target spectrum. The smaller the angle between each spectrum, the more similar the two pixels are. One aspect of SAM is insensitivity to changes in pixel illumination because increasing or decreasing intensity doesn't change the direction of the spectral vector, only the magnitude. In the higher dimensional case, hyperspectral pixels will form a hyper-angle that follows the same principle of smaller angle, better match [Shippert:2006].

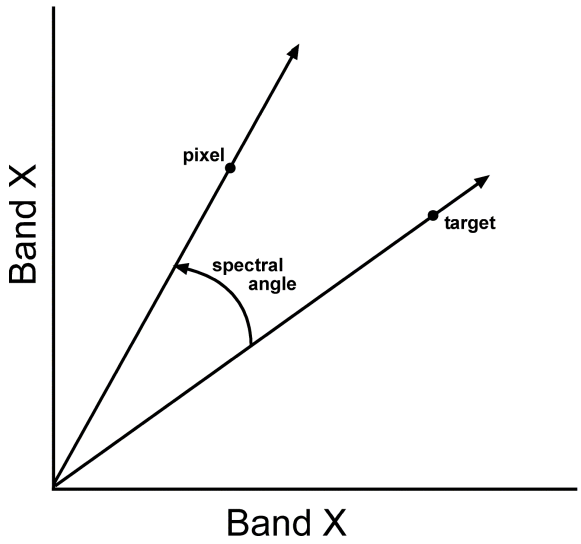


FIGURE 2.4 - The Spectral Angle Mapper (SAM) [Shippert:2006].

Depending on the results of using SAM to produce initial similarity scores, it might be advantageous to apply other statistical spectral matching techniques such as root mean squared error (RMSE). One limitation of SAM is the insensitivity to magnitude—although the mean spectral vectors may line up, a significant deviation in magnitude would not be detected. RMSE, on the other hand, is a simple statistical measure of the band-by-band deviation of each mean candidate spectra as compared to the mean target spectra, as seen in equation 2.3 [Taylor:1997],

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\bar{t}_i - \bar{c}_i)^2} \quad . \tag{2.3}$$

In the above equation, $\bar{t}_i$ is the i^{th} band of the target mean spectrum and $\bar{c}_i$ is the i^{th} band of the candidate mean spectrum. Each candidate would be measured against the mean spectrum of the current target.

Fundamentally, we can consider the object association step as a continuous target detection task, where a time-history of the spectral "signature" of each object—which can now be considered as targets—was provided in the previous frames. As a sufficient number of frames are accumulated, a running average or median of each spectrum could provide the best target signatures for object association in the next frame.

2.3.3 Track Management

Once the object association (or object matching) task has been accomplished, the surveillance system must decide how to label each object—whether it belongs to an existing track or becomes a new track. Although a full-blown tracking system is was not developed for this project, it is essential to understand the basic elements of a tracking system and how they relate to the detection, segmentation, and object association sub-tasks.

Many tracking systems are based on Kalman filters in order to predict the state of an object in the next frame. However, Kalman filter approaches are limited because they assume a unimodal Gaussian density that cannot support multiple motion hypotheses [Collins:2000]. The Kalman filter is also limited in some applications where a nonlinear motion model is required. The so-called Extended Kalman Filter (EKF) is an extension of the linear Kalman filter, applicable to nonlinear measurements and/or nonlinear target dynamics [Blackman:1999]. Multiple Hypothesis Trackers (MHT) were introduced by [Reid:1979] to form and manage multiple hypotheses whenever there are observation-to-track conflicts. It consists of a deferred decision logic in which alternative data association hypotheses are formed in anticipation that subsequent observations will resolve the conflict [Blackman:1999].

Another approach to Kalman filtering and MHT—as developed in the VSAM system for DARPA [Collins:2000]—is to maintain a list of multiple hypotheses to handle cases where object matching between multiple objects is ambiguous. The method assumes there are five tracking scenarios: 1) A new object appears; 2) An existing object disappears; 3) An object matches exactly one track; 4) A single track splits into multiple objects; or 5) Multiple objects merge into a single track. In this tracking system, object trajectories are also analyzed to reduce false alarms by evaluating object persistence and motion salience. The VSAM tracking method was determined to be suitable for this research, the details of which are covered in the methodology section.

The VSAM tracking system further divides the tasks into the following steps, which are covered in the following subsections:

- Predict positions of known objects
- Associate predicted objects with current objects
- Hypothesis Tracking
- Update object track models
- Reject false alarms

For a more complete understanding of a tracking system, the VSAM model provides an excellent example of how object association is integrated with track management, as described in the following subsections.

2.3.3.1 Predict Positions

The first step, predicting the location of objects in each frame, requires an estimate and uncertainty of the future position of each object being tracked. Given the time between frames Δt, the estimated position is simply based on the expected displacement due to the previous velocity estimate (assuming constant acceleration), as shown in equation 2.4 [Collins:2000],

$$p_{n+1} = p_n + \vec{v}_n \Delta t \quad . \tag{2.4}$$

Thus, the uncertainty in the position is based on the uncertainty in the velocity estimate used, as shown in equation 2.5 [Collins:2000],

$$\delta p_{n+1} = \delta p_n + \delta \vec{v}_n \Delta t \quad . \tag{2.5}$$

The estimated position is used to choose candidate moving regions in the current frame by extrapolating the previous location. The future position is then assumed to be somewhere between the previous location and a reasonable location within the bounds of what is physically possible. Keeping low frame rates in mind, thus considering a significant increment of time between frames, the object could conceivably change or even reverse direction. Thus, a ring based upon a maximum velocity is drawn around the previous location, with the most likely location being along the previous course. Any candidate object that falls within this ring will be considered for matching, with greater confidence given to a match on the expected course.

2.3.3.2 Associate Predicted Objects

Various object association techniques were discussed previously in section [2.3.1]. Because the focus of this research is to enhance existing methods by including spectral information, the tracking process described thus far is suitable for track management. However, the key concept of enhancing the object association step using spectral information is described in greater detail in the methodology section.

2.3.3.3 Hypothesis Tracking

As stated above, the crux of a tracking algorithm is in matching the existing tracks in the previous frame to the detected moving regions in the current frame. The results of the object association step can fall into one of five categories:

1) Existing track matches exactly one candidate – Best case
2) Existing track does not match any candidates – Stopped or lost
3) Existing track matches multiple candidates – Split
4) Candidate matches multiple existing tracks – Merge or Occlusion
5) Candidate does not match any existing tracks – New track

In the first case, where the existing object matches exactly one candidate, the update is simple. The state parameters (position, velocity, etc) are updated based on the matched object and the confidence score is increased. The second case—in which the object has stopped, been occluded, or left the scene—requires more information. Thus, the state parameters remain unchanged aside from a reduced confidence score until a future match is made. If the confidence score falls below an empirical threshold, the object track is terminated. The third case, where multiple candidates are reasonably close matches, can result from an object splitting into several objects, either in reality (such as passengers dismounting a vehicle) or due to a failure in the detection algorithm to properly cluster all the moving pixels into one object. In this case, the best match is assigned to the existing track with increased confidence. The remaining matches are considered new tracks with associated low confidence, pending further information. The fourth case is the alternative to the third case, where one candidate matches multiple existing tracks reasonably well. In this case, multiple moving objects have actually merged into one (such as passengers mounting a vehicle), are traveling near or are occluding one another, or the anomalous detection of multiple objects has been rectified. Further information is again required to determine what is actually happening. In this special case, the objects are tracked separately under the assumption they are most likely sharing the same region but not actually merged. Each existing track is updated using the same matching candidate object. If the multiple tracks continue along the same trajectory with the same velocity for a period of time they can be merged. Otherwise, they are tracked separately under the assumption they will split again in future observations. In the fifth and final case, a new object is hypothesized with low confidence pending further matches.

It is instructive to note that at the beginning of a tracking session—when no matches or tracks have yet been established—all detected moving objects are hypothesized as new tracks with low confidence and are equally likely to head in any direction. It is here, once again, that adding spectral information could provide an advantage over the traditional single-band tracker. With no position or velocity estimates, the matching process is weighted more heavily on spectral similarity. In comparing spatial confidence with spectral confidence, it is important to note that spectral confidence should *increase* with the number of bands (assuming the spectral bands are sufficiently uncorrelated). Conversely, spatial confidence should *decrease* with decreasing frame rate, because the uncertainty in the predicted location goes up as the time increment between frames increases.

2.3.3.4 Update Tracks

After all hypotheses have been established, track parameters are updated based on the matched objects. The updated position p_{n+1} is taken as the current centroid of the bounding box. The new velocity v_{n+1} is estimated using a weighted average of the newest velocity estimate $\hat{v}_{n+1}$

(displacement divided by Δt) and the previous velocity v_n, as shown in equation 2.6 [Collins:2000],

$$\vec{v}_{n+1} = \alpha\,\hat{v}_{n+1} + (1-\alpha)\,\vec{v}_n \quad , \tag{2.6}$$

where α is a time constant specifying how frequently old observations are updated. Likewise, the velocity uncertainty is updated as shown in equation 2.7 [Collins:2000],

$$\delta\vec{v}_{n+1} = \alpha\,|\vec{v}_{n+1} - \hat{v}_{n+1}| + (1-\alpha)\,\vec{v}_n \quad . \tag{2.7}$$

The spectral and grayscale templates of the object are updated in similar fashion using a running average (or median). However, in the case of multiple tracks being matched to a single candidate region (merge), the templates are not updated in order to preserve the previous individual appearance of the tracked objects. Again, this applies the assumption that the most likely scenario is objects traveling near one another that will eventually split again. Any track that has not been matched will not be updated except for a reduced confidence score. An object that has been tracked for several frames will have a relatively high confidence. In the event that an object temporarily stops or is occluded in motion, the track will persist for a number of frames before it is terminated. As such, a high confidence track will have a likelihood of being reacquired in a later frame.

2.3.3.5 Reject False Alarms

Possibly the most troublesome aspect of any detection and tracking system is the problem of false alarms. In order to validate a moving object as a legitimate target, a history must be established. There are two attributes that can validate such a target: persistence and motion salience. The persistence of an object is handled by the tracker in the form of an updated confidence score. Once an object falls below a confidence threshold the track is terminated. Thus, a newly detected object is given a low confidence and will remain above the threshold only if immediate updates increase the confidence. In contrast to new tracks, existing tracks with sufficient history will be allowed to persist longer without a match in anticipation of reacquisition of a valid object.

The second attribute, motion salience, was introduced in the background section [2.1.2.2] as an effective means of filtering out distracting, unimportant motion. Defined as directionally consistent motion, motion salience was first evaluated based on optical flow [Wixson:2000]. The history of a moving object can be accumulated over several frames giving the object a measure of salience; if the direction of flow changes, the salience measure is set to zero. However, optical flow is computationally expensive, overly complex, and not extremely accurate

when propagated over several frames [Adams:2006]. A short cut method [Collins:2000] provides a means to compute motion salience based on three parameters, which are initially set to zero: frame count c, cumulative flow d_{sum} and maximum flow d_{max}. For each frame, the displacement of each object is accumulated into d_{sum} and the frame count is incremented. If accumulated flow d_{sum} is greater than d_{max}, it is reassigned as the new maximum. If d_{sum} falls below 90% of the maximum flow d_{max}, it is assumed that the direction of the object has reversed and all parameters are set back to zero. The calculations are performed in both the x- and y-image direction in such a way that objects maintaining an accumulated displacement in either direction are considered salient, and thus valid targets.

In summary, the example tracking system described above will predict object locations, associate objects, manage multiple hypotheses, update valid tracks, and reject false alarms. However, this is simply one technique among numerous other tracking systems. For the purposes of this trade study, any such tracking system would be customized to meet user requirements. An essential step in modifying the tracking process is to enhance the object association step by taking advantage of spectral information. Once all moving objects are detected and associated, track management provides the time history and predicted state of each object. However, up to this point all detection and tracking techniques discussed were developed primarily for single-band video at 30 fps. Current research has branched into multi-sensor fusion (applicable to multispectral data) and tracking at lower frame rates.

2.4 Current Research

Two active areas of research that are pertinent to this trade study are visible/IR data fusion and low frame rate tracking systems. In either case, the research goal is to enhance motion detection and tracking performance. However, the two ideas are being looked at independently, without the intention of using data-fusion to enable low frame rate tracking. Research regarding fusion of visible and IR sensors has primarily been with the goal of either enhancing daylight surveillance or enabling nighttime surveillance. On the other hand, low frame rate tracking techniques have the exclusive goal of reducing collection bandwidth and/or enabling surveillance with very low cost equipment. An exhaustive review of the most current tracking technology shows very few people are approaching the problem in a multispectral sense. Up to now, the notion that multispectral data might enhance low frame rate tracking performance is apparently a unique one.

2.4.1 Visible/IR Fusion

Recent work on fusion of visible and infrared (IR) data attempts to leverage the combined benefits of using different modalities while compensating for failures in the individual modalities [O'Conaire_1:2006 – Comparison of Fusion Methods]. Using an appearance-based tracking method, one study compares fusion methods based on combining frame-to-frame similarity scores from individual modalities. The study is based on an adaptive appearance model using a mixture of Gaussian distributions to model each pixel [Zhou:2004]. However, revising this method, the adaptive model uses a single multidimensional Gaussian distribution for each pixel,

with a per-pixel importance weighting to track image regions. The study concludes that the most promising combination of individual trackers results from a simple multiplication of the similarity scores, called the similarity score product. Of particular interest, this study models the appearance of the object being tracked as a rectangular grid of *d* pixels, with each pixel being modeled by a Gaussian distribution. In this way, each pixel at a given time *t* can be represented by the mean vector of *k* values, where *k* is the number of features. Assuming pixel features are independent, the model also includes the diagonal covariance matrix to characterize each pixel. Such an arrangement is particularly attractive when considering the problem as a multispectral one, where each pixel is, in fact, *k*-dimensional. Finally, they include a weighting factor to each pixel that will remove background pixels from the object region while emphasizing valid features.

The above fusion approach is extended to track objects by using multiple spatiograms trackers [O'Conaire_2:2006]. In this way, the system can process *K* different channels of data by comparing *K* one-dimensional histograms. However, in the case of histograms the assumption of independence does not hold. The problem is resolved by introducing second-order spatiograms, which include spatial information by weighting each histogram bin by the mean and covariance of the pixel locations that contribute to that bin [Birchfield:2005 – Spatiograms versus histograms]. As a result, the technique successfully incorporates the results from each single-band tracker by using the combined product of the individual spatiogram similarity scores, as shown in equation 2.8,

$$\textit{Combined Similarity: } \rho(y) = \rho^{(1)}(y)\,\rho^{(2)}(y)\ldots\,\rho^{(K)}(y). \tag{2.8}$$

The technique of multiplying individual similarity scores for each separate spectral band to achieve a combined score is intuitive when considering each similarity score as an individual probability.

However, as will be explained in detail in the methodology section, it is more straightforward to match objects by first using a spectral similarity measure, followed by a single-band similarity score rather than combining *K* separate trackers. After a spectral comparison, the spectra could be reduced to a grayscale and then compared spatially. This is of particular concern when considering low frame rate data where significant time may have passed between frames. In this case, the spectral similarity might be more reliable and therefore weighted more heavily than the less certain spatial qualities of the target. Tracking objects at low frame rates has other potential pitfalls, as described in the next section.

2.4.2 Low Frame Rates

The challenge of tracking objects at a low frame rate—where the time between frames could be significantly longer than at video frame rates—falls somewhere between change detection and motion detection. However, even at one frame per minute not much has changed environmentally, assuming similar solar angles and atmospheric conditions. Likewise, targets

such as people and vehicles remain spectrally constant—a yellow school bus is still yellow and people and vehicles remain relatively the same temperature. Now the challenge becomes handling changes in motion, such as abrupt changes in direction and velocity that were not a problem at 30 fps. Another way of looking at the problem is to compare low frame rates to increased object velocity (or frame-to-frame displacement). In this case we can see that motion models will fail as uncertainty between observations increases. Successful matching then relies upon the spectral signature of targets remaining constant enough to find and associate every moving object from one frame to the next.

The most current work (in a notably sparse area of research), approaches low frame rate tracking as a means to improve processing time and to reduce bandwidth and storage limits [Porikli:2005]. The paper shows the anticipated degradation in tracking performance as a function of reduced frame rate, as seen in figure 2.5. A system tracking a single object tended to suffer the least degradation, while tracking multiple objects suffered the most due to object ambiguity.

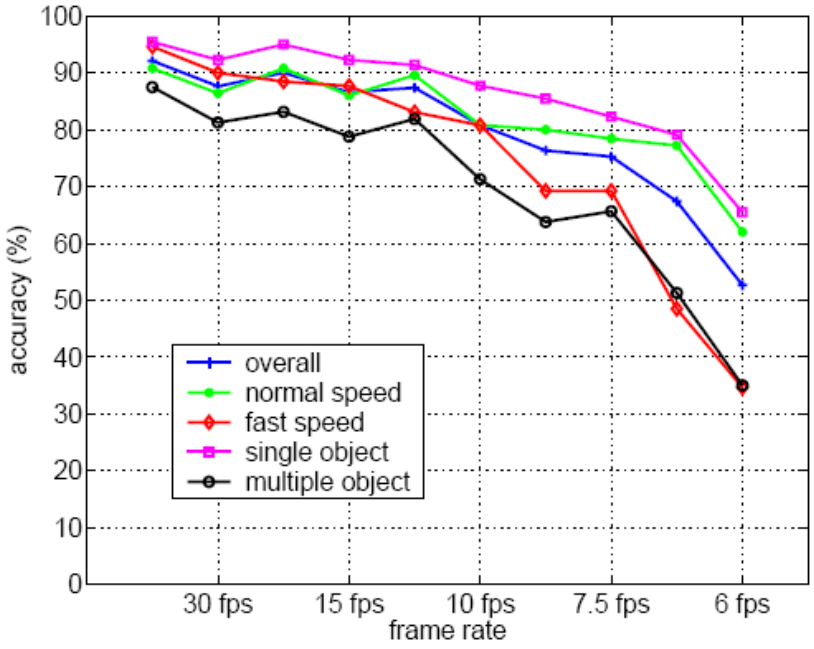


FIGURE 2.5 – Object Tracking Accuracy as a function of frame rate [Porikli:2005].

The tracking method used to generate this figure assumed object locations would overlap frame-to-frame, which is a legitimate claim using video at 30 fps. However, at lower frame rates there is an inherent flaw in that assumption because objects may have moved significantly between frames and thus are not overlapping. The solution is to start the object matching search at more than just the previous location. Multiple search starting locations are provided by the motion detection results whereby areas of motion indicate candidate matches for the target. Although this technique is effective in capturing all candidates regardless of spatial distribution,

it does not attempt to add spectral information to help determine the otherwise ambiguous location of objects in the next frame. The technique of attempting to associate targets with all candidates—regardless of location—will be addressed in the methodology section.

Considerable progress is being made in both data fusion and low frame rate trackers. However, the intention of this project was not necessarily to solve either of these problems independently. Rather, it is to investigate trades in performance as a function of the number of spectral bands and frame rates simultaneously. As such, performance metrics for moving object detection and association are needed to compare results using different settings within this trade space.

2.5 Performance Metrics

As presented by [Bashir:2006], there are two basic methods to evaluate tracking system performance by using either frame based or object based metrics. To achieve an overall perspective of the trade space for this project, the performance metrics derived for this study were somewhat less complex. However, frame- and object-based metrics are included in this section for future consideration when evaluating a complete tracking system. In addition, the perceptual complexity of a scene can be useful when evaluating the performance of a system [Black:2003].

2.5.1 Frame Based Metrics

In frame based metrics, each frame is evaluated individually for agreement between system results and the ground truth (GT) map for that frame. In this case, when comparing a system frame to the ground truth frame, two object bounding boxes are "coincident" if one centroid lies within the other box. Once each frame is evaluated, there are a number of metrics that can be considered by computing the total number of frames that meet each of the following criteria [Black:2003]:

- **True Negative (TN):** System agrees with GT on absence of an object
- **True Positive (TP):** System agrees with GT on presence of an object
- **False Negative (FN):** System does not report object when GT does
- **False Positive (FP):** System reports object when GT does not
- **Total Ground Truth (TG):** Total number of frames with ground truth objects
- **Total Frames (TF):** Total number of frames in video sequence

These are the fundamental measurements used in computing basic performance metrics as shown in equations 2.9 through 2.11. Once all of the above quantities are calculated for all the frames in the video sequence, the following can be computed [Black:2003]:

$$\text{Tracker Detection Rate (TRDR)} = \frac{\text{TP}}{\text{TG}}$$

$$\text{False Alarm Rate (FAR)} = \frac{\text{FP}}{\text{TP+FP}}$$

$$\text{Detection Rate} = \frac{\text{TP}}{\text{TP+FN}} \qquad (2.9\text{-}2.11)$$

The tracker detection rate (TRDR) and false alarm rate (FAR) characterize the tracking performance of the object-matching algorithm, while the detection rate (DR) indicates the tracking completeness of a specific ground truth track. As an example of how these metrics can be used to compare tracking systems, figure 2.6 shows TRDR and FAR results for six different tracker/detector combinations [Black:2003]. A similar comparison could be made between a system using a variable number of bands and/or variable frame rates.

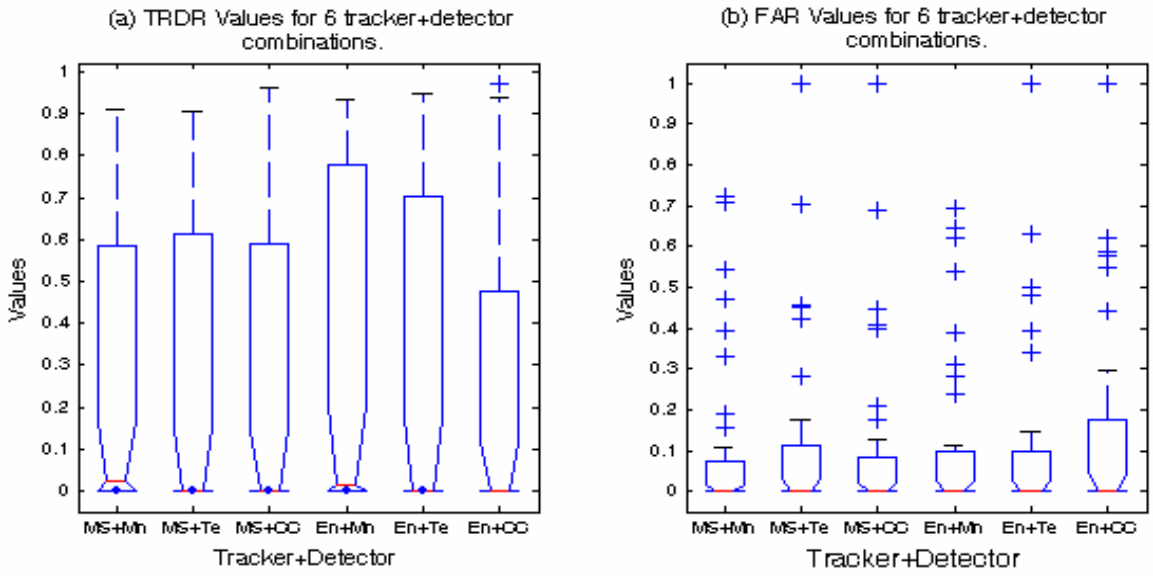


Figure 2.6 – (a) TRDR and (b) FAR for six combinations of trackers and detectors [Black:2003].

Other more specific metrics, as presented by [Bashir:2006], can be generated as shown in equations 2.12 through 2.17:

$$\text{Specificity} = \frac{TN}{FP+TN}$$

$$\text{Accuracy} = \frac{TP+TN}{TF}$$

$$\text{Positive Prediction} = \frac{TP}{TP+FP}$$

$$\text{Negative Prediction} = \frac{TN}{FN+TN}$$

$$\text{False Negative Rate} = \frac{FN}{FN+TP}$$

$$\text{False Positive Rate} = \frac{FP}{FP+TN} \qquad (2.12\text{-}2.17)$$

The above metrics are based upon counting the number of frames that either agree or do not agree with the ground truth frames and then considering the desired ratio to the total number of relevant frames in the video sequence.

2.5.2 Object Based Metrics

In contrast to frame based methods, object based metrics evaluate each object over the entire track. These metrics are based on a simple threshold-based correspondence. For each common frame between a system track (TR) and ground truth (GT) track, the Euclidean distance between their centroids is computed. The cumulative Euclidean distance is then normalized by the total number of overlapping frames between the GT/TR pair being evaluated. Finally, two GT/TR pairs are declared corresponding if their total normalized distance is within a threshold. Figure 2.7 shows the definitions of these four metrics (TN, TP, FN, and FP) over an entire sequence [Bashir:2006]. Notice that metrics for multiple objects in a single frame are computed. A single GT track could correspond to more than one TR, thus a correspondence map can be established based on the threshold.

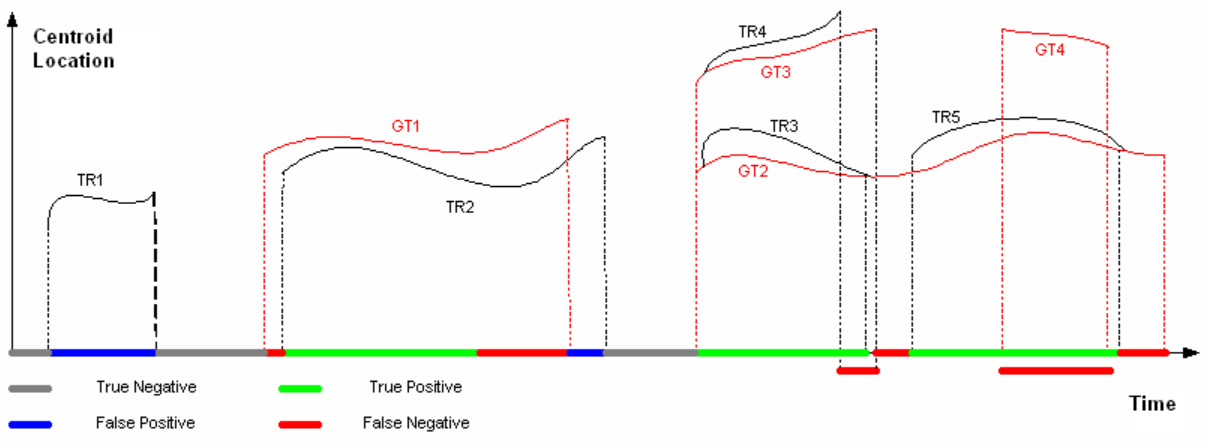


FIGURE 2.7 - Object Correspondence Map [Bashir:2006].

Once a correspondence is established, the true positive (TP), false positive (FP), and total ground truth (GT) are computed as explained in the frame-based method. The tracker detection rate (TRDR) and false alarm rate (FAR) are likewise computed similar to frame based methods.

Finally, a single-value called the object tracking error (*OTE*) can be computed as the average discrepancy between the GT bounding box centroid and the system result centroid, as shown in equation 2.18,

$$OTE = \frac{1}{N_{rg}} \sum_{i \in g(t_i) \wedge r(t_i)} \sqrt{\left(x_i^g - x_i^r\right)^2 + \left(y_i^g - y_i^r\right)^2} \quad . \tag{2.18}$$

In this computation, N_{rg} is the total number of overlapping frames between ground truth and system results over the entire video sequence. Ground truth coordinates (x_i^g, y_i^g) and system result coordinates (x_i^r, y_i^r) are the respective image locations of the object centroids, where i-subscripts indicate the i^{th} frame. In this way, a single performance score can be assigned to each object tracked by the system. A combined score would simply be the combination of an *OTE* for each object in the sequence. Although a single score for each tracking system configuration is useful, the combined score might be oversimplified unless the complexity of the tracking scenario is also considered.

2.5.3 Perceptual Complexity

As described by [Black:2003], the perceptual complexity of a scene can be controlled by a set of tunable parameters using “pseudo-synthetic” video sequences. The two parameters suggested are the maximum number of objects to be tracked (*MAX*) and the probability of creating a new object (*PN*), given that *MAX* has not been exceeded. In this sense, the pseudo-synthetic scene generation allows for a variable number of objects to be created and tracked. An example is

shown in figure 2.8, where perceptual complexity (average number of objects per frame) increases with the probability of adding a new object in any given frame [Black:2003].

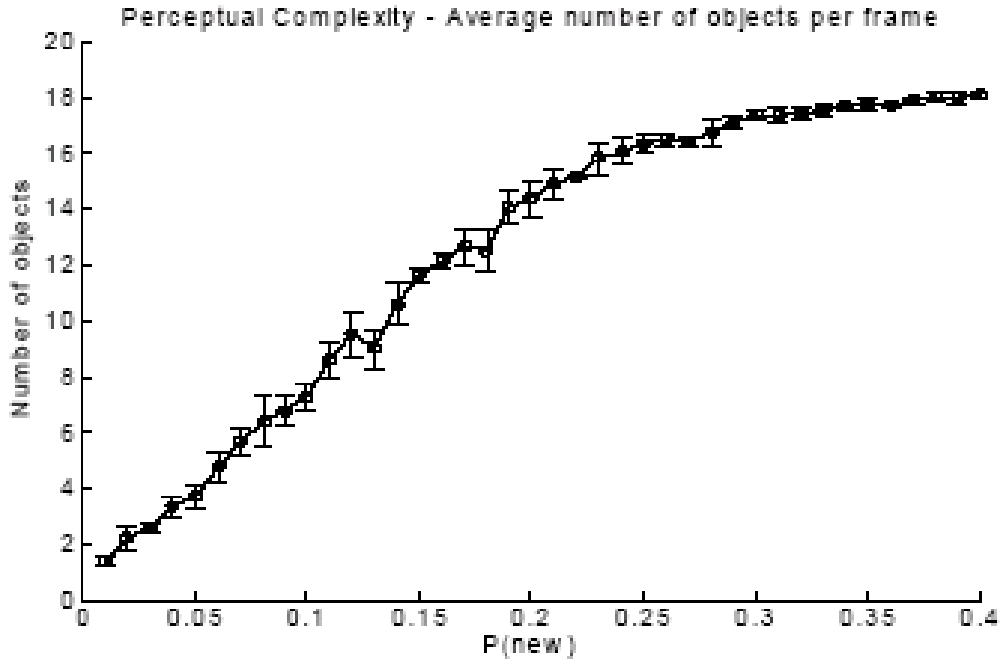


FIGURE 2.8 – Perceptual Complexity [Black:2003].

Although the datasets for this project did not allow for a variable probability of new objects, the synthetic video sequence was designed to have an increasing number of targets as a function of time. This allowed for a sense of perceptual complexity in that early scenes are simple and become more complex as new objects are added.

2.6 Background Summary

The above background section has isolated a few of the best moving object detection, association, and tracking performance measuring techniques from the wide body of literature on the subject. Based on the desire to increase spectral resolution and reduce frame rate, suitable existing algorithms have been selected. By enhancing spatiotemporal texture vectors with additional spectral detail—combined with applying spectral similarity to segmentation and object association—a novel approach was developed with the potential for improving system performance at reduced frame rates. This trade study investigates a new methodology in detecting, segmenting, and associating moving objects. The trade space includes test metrics, datasets (both synthetic and real world), and motion truth – as detailed in the next section.

Chapter 3

Methodology

After reviewing the state-of-the-art in moving object detection and tracking systems, a hybrid approach was devised. The newly devised object detection and association system modified current algorithms by including spectral information with the specific goal of achieving improved performance at reduced frame rates. The system model shown in the introduction (figure 1.3) provides a framework to determine which subtasks might gain by adding spectral information, and to what degree. The emphasis was on moving object detection, segmentation, and association. Once implemented, a trade study was conducted to determine system performance as a function of spectral (number of bands) and temporal (frame rate) resolution. In order to perform such a trade study, performance metrics were needed. Both synthetic and real world datasets were generated to provide relevant results and conclusions.

In order to limit the scope and objectives of this project, some simplifying assumptions were made. The scene collection was assumed to be from a stationary platform over an urban environment; thus, parallax and platform stability issues were not addressed. With an emphasis on developing a theoretical methodology, processing power and data storage were not considered limited by any specific system requirements. Finally, all frames were assumed to be registered to less than one pixel accuracy, which was perfectly true with synthetic data.

Because this was the first phase for the Center of Imaging Science to investigate multispectral motion detection and tracking, the emphasis was on daytime scenes in the visible through infrared regime. However, the methodology developed is intended to be extendable to future projects using only thermal bands for nighttime and/or low-light operations. As a final caveat, the proposed project was not to build an end-to-end tracking system. The emphasis was to determine the effects of spectral and temporal resolution on the motion detection, segmentation, and object association subtasks. Therefore, state vector predictions (i.e. position and velocity) were not available for object association. The intent was to measure detection,

segmentation and object matching performance under the hypothesis that improved performance in these subtasks is directly correlated to improved performance in a complete tracking system.

3.1 System Model

Starting with the model of a surveillance system, the intention of this research was to focus on potential improvements in a subset of tasks. As highlighted in figure 3.1, modifications to existing moving object detection, segmentation, and association techniques were studied. The first step was to modify single-band spatiotemporal texture vectors [Miezanko:2006] to include additional bands of data. Hence, a more sensitive detector was expected. The second step, segmentation, also gained an advantage by discriminating between background pixels and clustering spectrally similar pixels into objects. The third step, object association, measured the effective advantage of a multispectral system in matching an object from one frame to the next.

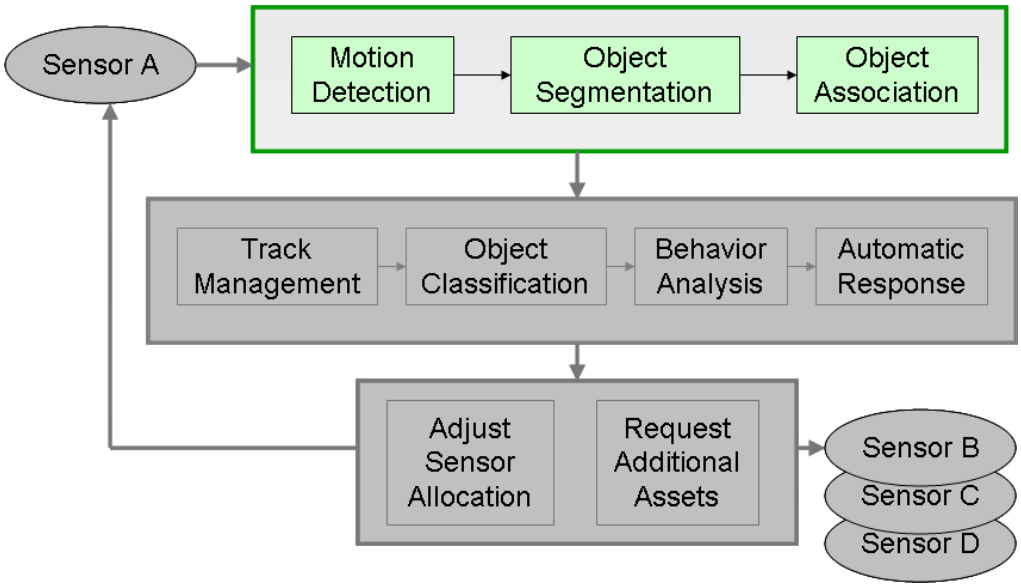


FIGURE 3.1 – System Model with subset of tasks highlighted.

3.1.1 Motion Detection Using Spatiotemporal Texture Vectors

As described in the background section [2.1.3], a video sequence can be represented by a set of spatiotemporal texture (SP) vectors. A flow diagram of the detection process is seen in figure 3.2, with SP-vectors as the input and detection motion-matrix as the final output. The steps in between reduce dimensionality of the SP-vectors, detect motion using maximum temporal variability, and threshold a motion measure to tag blocks of pixels as moving or stationary.

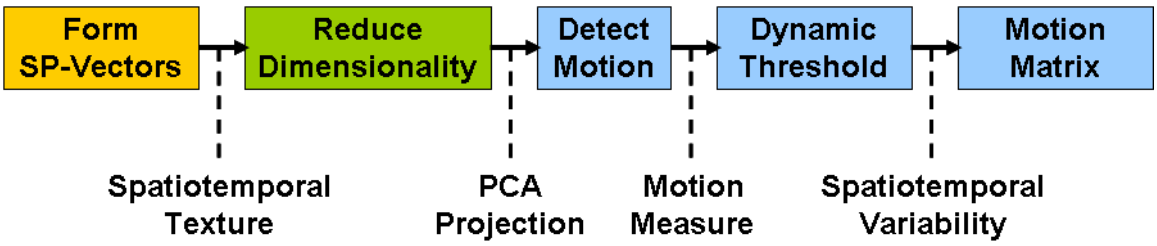


FIGURE 3.2 – Motion Detection Flow diagram.

The accumulation of pixel values over space and time causes the SP-vectors to have high dimensionality, which is exacerbated by adding spectral bands. However, dimensionality was reduced—even in the single-band mode—by using principal component analysis (PCA). PCA projections used an estimated covariance matrix based on an initialization period where it was assumed there is little or no motion in the scene. Once the SP-vectors were reduced to a manageable size, temporal variability was monitored as compared to the initial state where no motion was assumed. A motion measure was then produced by taking the largest eigenvalue of these dimensionally-reduced SP-vectors as accumulated over a sliding temporal window. In this way, a single value was assigned to each two dimensional region, or "block" (a subspace of the entire scene), at a given time. Next, a dynamic threshold then determined if the variability in that local spatiotemporal region indicated motion. The motion detection results for all blocks over all image frames are captured in the motion-matrix. These five steps are described in detail in the following subsections.

3.1.1.1 Formation of Texture Vectors

The first step is to divide each image frame into a grid of two dimensional square regions (blocks), whereby each block of the entire frame is monitored individually. Although (8 x 8) blocks of pixels were used in the original paper, the author indicated that regions of (4 x 4) pixels work just as well to determine local variability. As such, smaller blocks provide better spatial resolution [Miezianko_Notes:2007]. In this case, the datasets were derived from single-band campus security video cameras. Of course, the division of frames into a particular grid is dictated by the resolution of the sensor and/or dataset. Based on the hypothesis that additional spectral texture would improve sensitivity, (2 x 2) blocks—or even individual pixels—might also be monitored. However, computational performance decreases as the size of each spatial window is reduced because more blocks per frame have to be processed in order to monitor the entire scene.

Next, each two dimensional region (or block) in the scene is accumulated over several frames, adding the third dimension of time. These three dimensional, spatiotemporal texture blocks are then structured into a vector, as seen in figure 3.3.

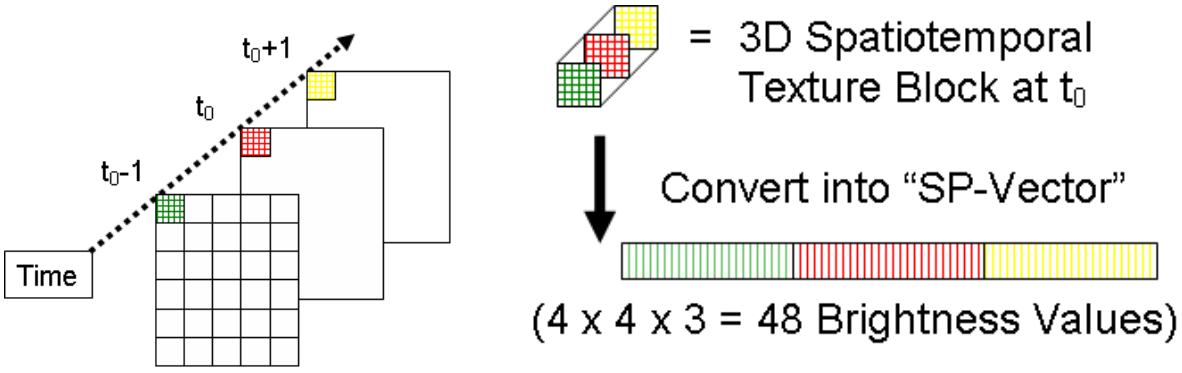


FIGURE 3.3 – Convert 3D block into SP-vector.

The example above shows an accumulation of (4 x 4) pixel blocks over 3 frames of grayscale video. The single-band setup produces SP-vectors with 48 brightness values. Now consider using a five-band system (as described in the datasets section) which produces SP-vectors with 240 elements. Keep in mind that the entire scene is divided into separate two dimensional blocks such that each SP-vector represents a single spatial subset over 3 frames. Thus, extending the spatiotemporal vector technique to multispectral data is simple and convenient. The single band SP-vectors are extended to a length, L_m as seen in equation 3.1,

$$L_m = (4 \text{ x } 4 \text{ x } 3 \text{ x } d). \tag{3.1}$$

The above equation uses (4 x 4) blocks over three frames, where d is the spectral dimension. However, it takes many such SP-vectors to represent the entire scene for a single frame.

3.1.1.2 Reduce Dimensionality

Fortunately, this technique is easily extended to multispectral data because the next step is to reduce the dimensionality of the SP-vectors, whether from a single-band or multi-band system. The first step in the PCA transformation is to zero-mean each SP-vector ($\boldsymbol{b}_{I,J,t}$), where $\boldsymbol{b}_{I,J,t}$ represents spatiotemporal texture at location (I,J) and time t. The principal component projection ($\boldsymbol{P}^{K}_{I,J}$) is computed for each location (I,J) over an initial period of time where little or no motion is assumed. Using the example of 4x4 pixel blocks over three frames, the 48-element SP-vectors generate a 48x48 covariance matrix. By using a value of $K=10$, we keep the first ten principal components to describe each SP-vector ($\boldsymbol{b}^{*}_{I,J,t}$), as seen in equation 3.2,

$$\boldsymbol{b}^{*}_{I,J,t} = \boldsymbol{P}^{K}_{I,J} \bullet \boldsymbol{b}_{I,J,t} \, . \tag{3.2}$$

Although the author used a value of K=10, the number of principle components was found to be dataset dependent. In fact, significantly more principle components were required for the WASPLITE data, as discussed in the results section. In order to maintain a current background model, $\boldsymbol{P}^{K}_{I,J}$ for each location can be periodically updated. These updates would only use observations where that location is determined to be stationary (i.e. no motion detected).

To better visualize the data structure, figure 3.4 shows the spatiotemporal texture vectors of a single location *(I, J)* over an entire video sequence, accumulated frame-by-frame. Given a total of N frames in the sequence, each row in the (N x 48) array represents the single-band texture vector at a given time. For example, the first row represents spatiotemporal texture at location *(I, J)* for time t_0. After the initialization frames (usually 50) are accumulated, $\boldsymbol{P}^{K}$ can be applied to each SP-vector, reducing the dataset to an (N x 10) array. When this technique is extended to five spectral bands, the resulting (N x 240) array can still be reduced to a limited number of principle components.

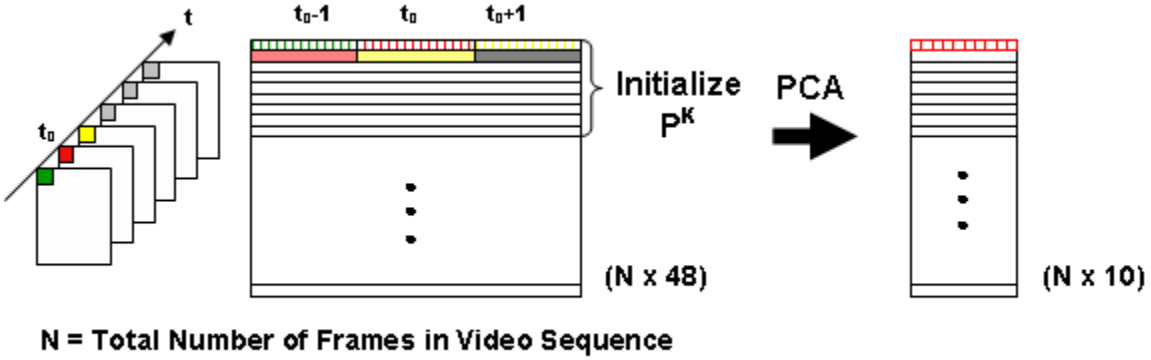


FIGURE 3.4 – PCA Reduces SP-vectors into 10-element vectors

Using the synthetic (DIRSIG) single-band data, keeping the top ten principle components preserved about 99% of the information content. However, using multispectral data brought up the question of whether or not this would be sufficient for the longer SP-vectors associated with higher dimensional data; especially relevant when using the real world (WASPLITE) data. An analysis was conducted on both synthetic and real motion data to determine the correct number of principle components, as explained in the results section.

Once a suitably defined SP-vector had been established for each spatial block over three temporal frames, the next objective was to determine if one particular spatial/temporal location should be labeled as moving or stationary.

3.1.1.3 Detect Motion Based on Temporal Variation

In order to detect motion at each location, the dimensionally reduced SP-vectors ($\boldsymbol{b}^{*}_{I,J,t}$) are monitored for temporal outliers. After the initialization period, incoming frames are grouped

into a symmetric sliding temporal window of W frames before and after the current frame (equation 3.3):

$$[\, b^*_{I,J,t-W}, \ldots , b^*_{I,J,t}, \ldots , b^*_{I,J,t+W} \,]. \tag{3.3}$$

Setting $W = 3$, a temporal window of seven frames is accumulated into a (7x10) array of texture values, as depicted in figure 3.5. However, note that using low frame rate data could allow for the number of temporal samples $(2W+1)$ to be adjusted.

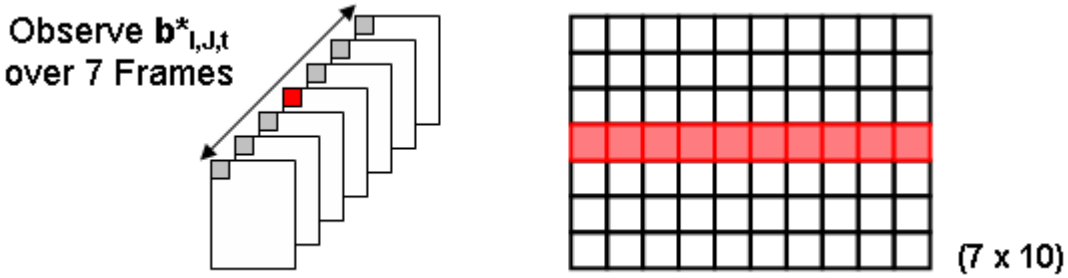


FIGURE 3.5 – Temporal Window to Compute Motion Measure (mm).

Next, a measure of motion is determined by computing the (10x10) covariance matrix of the (7x10) array of values (seven observations of a 10-element vector). Using eigenvalues again, this time to determine the magnitude of maximum variability in the sample set, the largest eigenvalue was assigned as the "motion measure" *(mm)*. Thus, *mm* is a function of both location *(x, y)* and time *(t)* within the video sequence, as shown in equation 3.4 [Miezanko:2006].

$$mm\,(x, y, t) = (\Lambda_{x, y, t})_{max} \tag{3.4}$$

Assigning the largest eigenvalue as the motion measure makes sense because the associated eigenvector represents the direction of maximum variation within the local spatiotemporal dataset. Thus, each two dimensional (4 x 4) pixel region at a given time t_0 is represented by a single motion measurement (or score). The entire scene is monitored by a number of these regions, depending on the dimensions of the frames.

For example, a (240 x 240) pixel frame would be divided into 3,600 separate (4 x 4) pixel windows for observation of motion across the scene. In order to monitor a sequence of frames over the entire time period, a sliding window in time is applied to evaluate each (4 x 4) region at a given time (i.e. one frame), as depicted in figure 3.6. A motion measure (mm_1) for the upper

left region at time t_1 is computed from the adjacent frames, as seen in red. Similarly, a new motion measure (mm_2) is computed for the same region at time t_2, as seen in green.

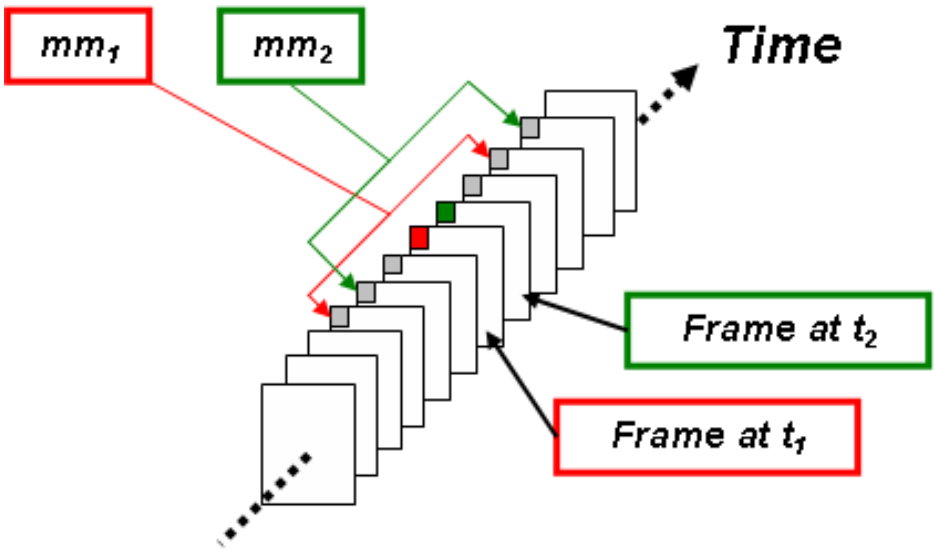


FIGURE 3.6 – Sliding Temporal Window.

Given a motion measure for each region of the current frame, the associated pixels are labeled either stationary or moving based on a dynamic threshold.

3.1.1.4 Dynamic Threshold

At any fixed position, *mm(x, y, t)* can be considered *f(t)*, a function of time only. To set the initial threshold, the *mean* and standard deviation (*std*) of *f(t)* are computed over the same initialization period used for the PCA. An outlier indicating motion is identified if it exceeds the threshold C_1, as described in equation 3.5 [Miezanko:2006].

$$\frac{f(t)-mean(t-1)}{std(t-1)} > C_1 \quad \text{where } C_1 \text{ is a constant and } std(t)=\sqrt{\sigma^2(t)} \tag{3.5}$$

In this case, the region is tagged as moving and the mean and standard deviation are no longer updated. If the motion measure *f(t)* falls below C_1 and remains above C_2, the region is not tagged as moving. However, the mean and standard deviation are still not updated until the value of *f(t)* falls below C_2, as described in equation 3.6 [Miezanko:2006].

$$\frac{f(t)-mean(t-1)}{std(t-1)} < C_2, \qquad C_2 < C_1 \tag{3.6}$$

When *f(t)* falls below C_2, it is once again considered stationary and the mean and standard deviation are updated. Thus, *mean(t)* and *std(t)* are dynamically updated only when outliers are not detected (i.e. the pixels are stationary). These values are computed based on a running average, as described in equations 3.7 through 3.9 [Miezanko:2006].

$$\begin{aligned} mean\,(f(t)) &= u \cdot mean\,(f(t-1)) + (1-u)\cdot f(t), \\ \sigma^2(f(t)) &= u \cdot \sigma^2(f(t-1)) + (1-u)\cdot (f(t) - mean\,(f(t-1)))^2, \\ std\,(f(t)) &= \sqrt{\sigma^2(f(t))}. \end{aligned} \tag{3.7 - 3.9}$$

Typical settings for the threshold variables from the original paper [Miezanko_Notes:2007] were (equation 3.10):

$$\begin{aligned} C_1 &= 50 \\ C_2 &= 10 \\ u &= 0.99. \end{aligned} \tag{3.10}$$

The constant *u* is the portion of the previous variance being retained in the update. However, these values were used for a specific set of data. The threshold settings used for this project were determined experimentally based on the characteristics of the synthetic and real world datasets, respectively. Ultimately, values for these threshold variables were selected to reduce missed detections at the expense of allowing more false alarms – consistent with the operational performance objective established from the beginning of this project.

The process of defining these values was a combination of trial and error and analysis, an example of which is shown in figure 3.7. In this example, the y-axis is the motion measure (*mm*, in blue) for one particular block over every frame in the video sequence. The frames where motion was detected have noticeably higher *mm* values than the background level. By adjusting C_1, C_2, and *u,* the logical detection results (in Green) can be compared to the motion truth detections (dashed magenta line). The logical yellow detections are the result of the original variable settings, to which the results of variable threshold values were compared. The difference in "magnitude" of the logical detections is simply to distinguish the different cases (whereas the *mm* values are to scale on the y-axis). Also, note the detections around frame 1600 do not agree with the motion truth (i.e. they are false alarms). Conversely, all of the experimental detections appear to overlap all of the motion truth detections. Again, the emphasis on setting these threshold variables was to reduce missed detections, at the expense of tolerating more false alarms.

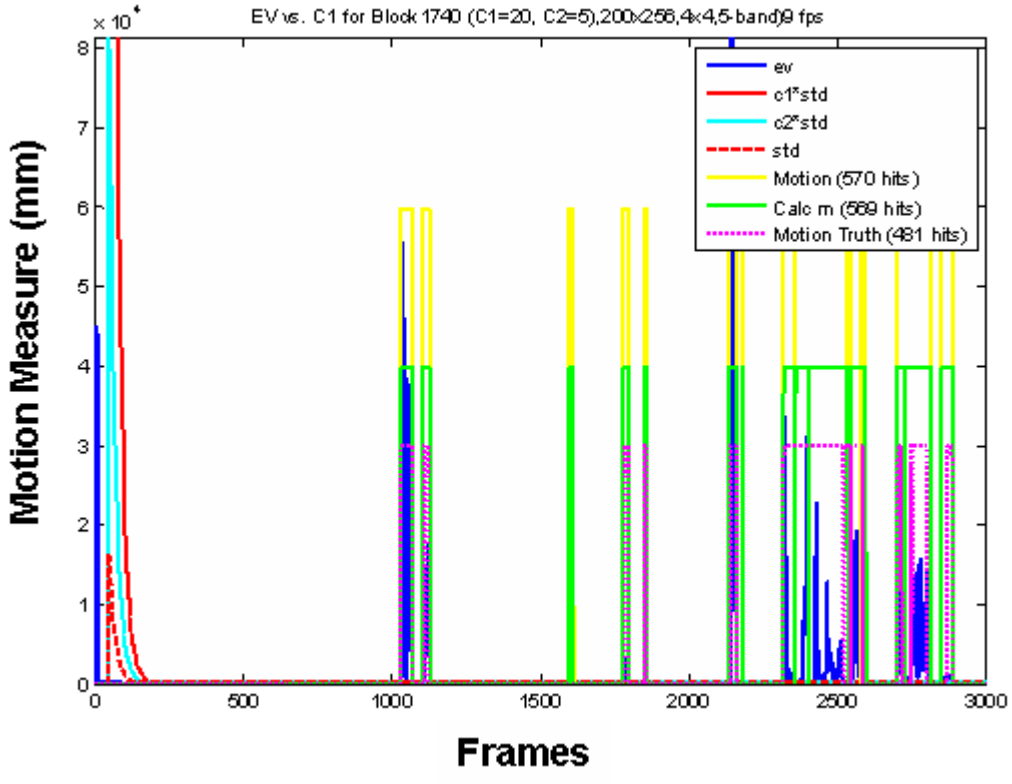


FIGURE 3.7 – Example of Detection Results for a Single Block.
(Method for Experimental Results Using Variable C_1, C_2, and u Values)

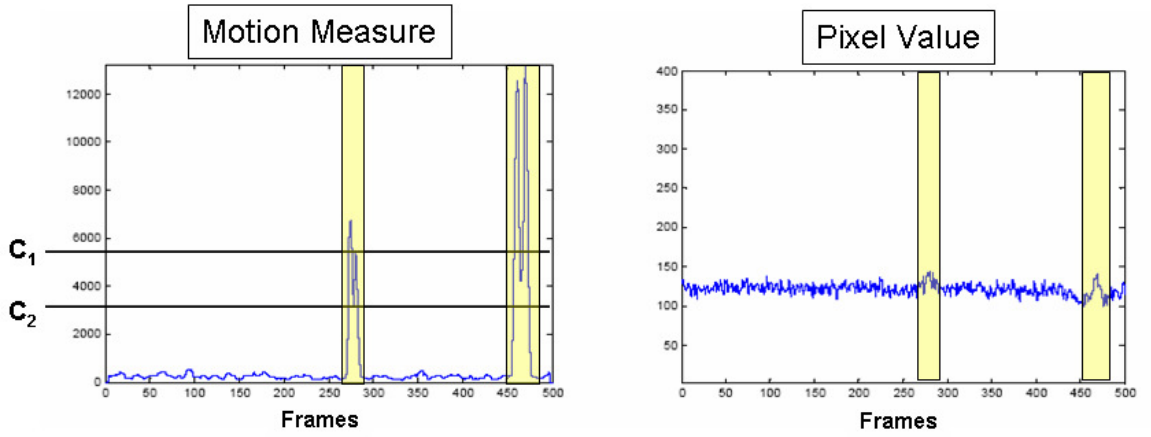


FIGURE 3.8 – Dynamic Threshold Example for a Single Block [Miezanko:2006].
(Method to Compare Motion Measure Technique to Raw Pixel Values)

To better understand these threshold settings, an example of variability at a given location is shown in figure 3.8 [Miezianko:2006]. Values for the motion measure, *mm*, can be plotted as a function of time (i.e. frame number), for a given spatial *(x, y)* location (figure 3.8, left). The motion measure clearly exceeds C_1 over two separate time intervals (highlighted). The pixels corresponding to that *(x, y)* location were labeled as moving as whenever *mm* exceeded the C_1 threshold. When *mm* fell below the C_1 threshold but remained above C_2, motion was not detected. However, *mean* and *std* were not being updated yet. Finally, when *mm* fell below C_2, *mean* and *std* updates resumed.

In contrast to the easily observed *mm* outliers (figure 3.8, left), the raw pixel values (figure 3.8, right) are also shown as a function of time. Notice that variability in pixel values was not considerably different during the same two highlighted time intervals. Thus, the motion measure provides a more distinct value with which to assess motion. Having identified motion at the pixel (or block) level, the entire spatial scene was evaluated in the same manner. The dynamic threshold process was then applied to the entire video sequence and assembled into a single result called the motion-matrix.

3.1.1.5 Motion Matrix

Once the dynamic threshold process identified all moving pixels in each frame, a motion-matrix was constructed to assemble all detection results into a single matrix. The motion-matrix is formed such that the number of frames is represented by columns (x-axis), while the motion detection results for each block fills the rows (y-axis). Using the example of a (256 x 256) pixel scene over 4,400 frames, an example of the resulting motion matrix can be seen in figure 3.9.

The motion-matrix gives an immediate sense of the amount of motion in a video sequence. The logical matrix shows motion wherever the value is equal to one, whereby the majority of background values (i.e. motionless) are equal to zero. A similar matrix can be constructed using truth data for comparison, as discussed in the datasets section. Another characteristic of the motion-matrix (figure 3.9) is that certain pixel blocks appear to be “stuck on” (i.e. moving) for many frames in a row. These apparent anomalies are seen as horizontal lines in the motion-matrix.

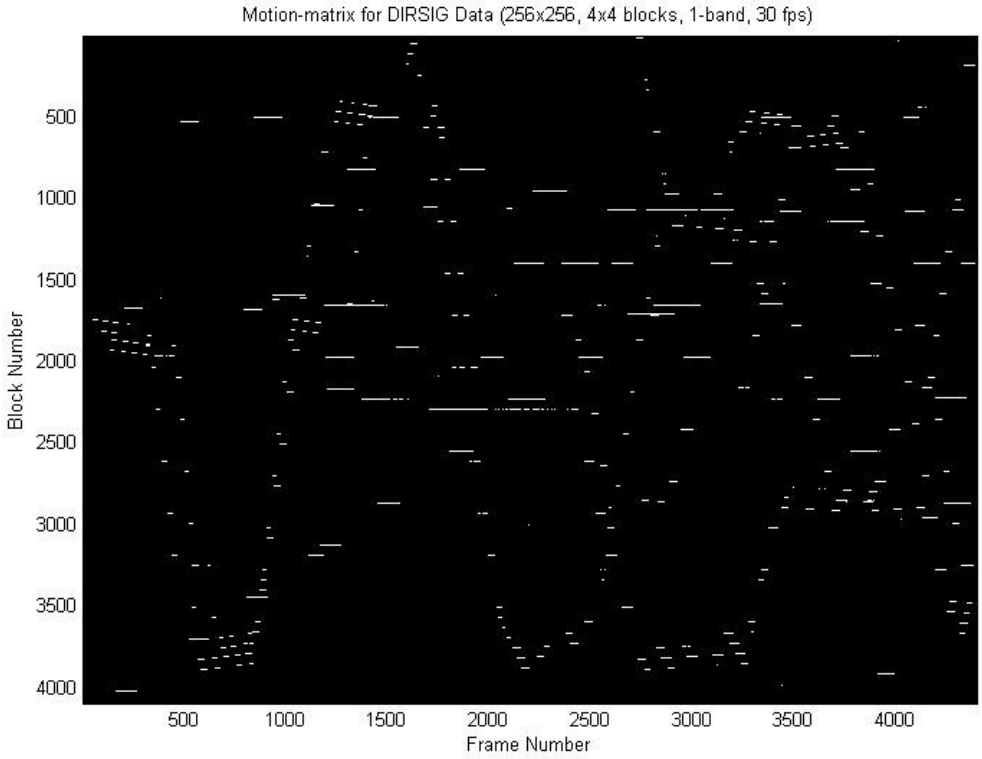


FIGURE 3.9 – Motion-matrix for a (256 x 256) Pixel Scene Over 4,400 Frames.

FIGURE 3.10 – Motion Detection "Ghosting" At Low Frame Rates.

Initial results from testing the single-band algorithm revealed these same anomalies, only more so, at reduced frame rates. At low frame rates (e.g. 3 fps vice 30 fps), anomalous detections are seen before and after the actual moving object appears (figure 3.10). This anomaly, here on referred to as "ghosting", can be interpreted either spatially or temporally. However, it is the direct result of relying on a seven-frame time sample to determine if temporal variability is sufficiently high enough to tag a block of pixels as moving or not.

As frame rate is reduced, moving objects can be interpreted as moving "faster". In the case of a small, fast moving object, the seven frame temporal window may only register the object in one of the seven frames. In this case, the sliding temporal window will falsely identify motion in all seven frames. Among the other advantages of multispectral data, a spectral filter prior to the segmentation process was used to address this problem.

3.1.2 Spectral Filter

In order to address the "ghosting" problem, a spectral filter was developed to check the detection results against a background model. The background model consists of the median of fifty previously observed frames. In this way, any outliers (i.e. moving objects) are eliminated from the current model, as seen in figure 3.11.

In this example, we have a moving car (top, circled in yellow) that has been removed from the background model (bottom). Thus, the background model, composed of previous frames, can be used to test potential ghost-detections. If the mean spectrum of the ghost-object is similar to the background model spectrum at that spatial location, it is not tagged as a legitimate moving object. Here the spectral filtering process begs the question: Why not use this technique to identify all moving objects? The answer is the spatiotemporal vector method works much better as a motion detector than simple background differencing. However, background differencing comes in handy for checking the results and eliminating ghosting due to low frame rate data. It is worth noting that multispectral data, once again, has an advantage over single-band data when making a spectral comparison, as will be seen in the results section.

As discussed in the previous section, "ghosting" becomes more apparent in the motion-matrix at low frame rates, as seen below in the example below (figure 3.12, top). After the spectral filter was applied to this data the motion-matrix is notably cleaner (figure 3.12, bottom) with less horizontal lines. This example demonstrates that anomalous detections can be removed if they are determined to be background pixels. Given this multispectral method of detecting moving blocks of pixels in each frame, the next task was to segment these blocks into moving objects by clustering spatial neighbors into spectrally consistent, discrete objects.

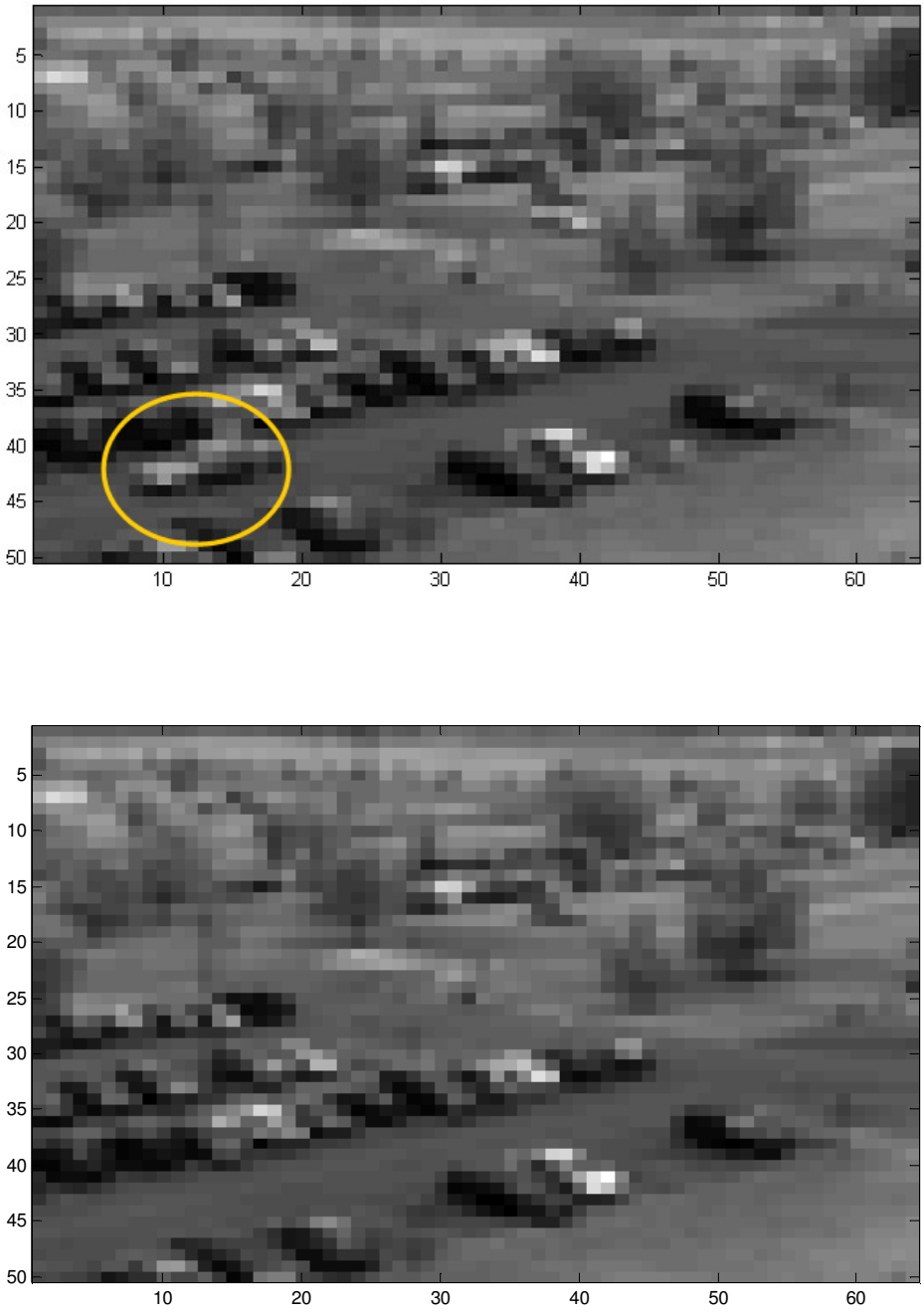


FIGURE 3.11 – WASPLITE Background Model.

Single Data Frame (Top), Background Model (Bottom)

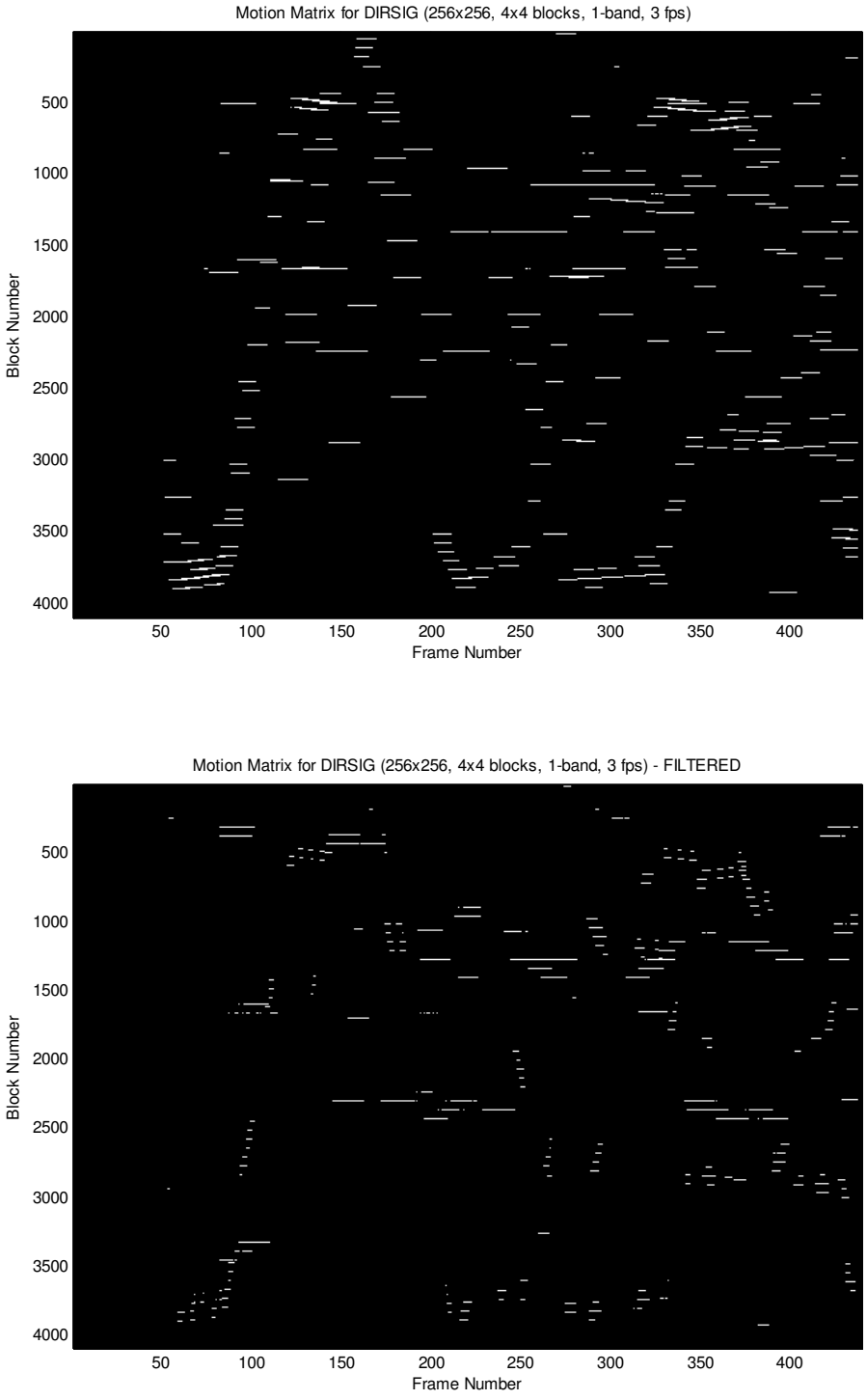


FIGURE 3.12 – Spectral Filter Applied to Motion Matrix (3 fps) .

Not Filtered (Top), Filtered (Bottom)

3.1.3 Object Segmentation

Once moving pixel regions were detected, neighboring regions were combined to define individual objects. Morphological processes such as connected components were applied to combine neighboring blocks where motion was detected [Gonzalez_Woods:2001]. Additionally, dilation and erosion techniques were used to better define the objects—more relevant to "real world" data than simulated datasets.

3.1.3.1 Overview of Segmentation Process

The segmentation of moving objects was a three-step process, as outlined in figure 3.13. First, a connected-components routine was used to combine neighboring blocks of pixels and number each object in the frame. Second, morphological functions (erosions and dilations) were applied to the connected components to remove speckle noise (detections of 1 block or less) and to refine each object as a "blob" of pixels. Third, each blob was given a bounding box and labeled with various attributes of the object, such as centroid coordinates, number of pixels (size of object), and so on. Most importantly, the labeling process also captured and saved the mean spectral vector of each blob in each frame, which could be used for object association.

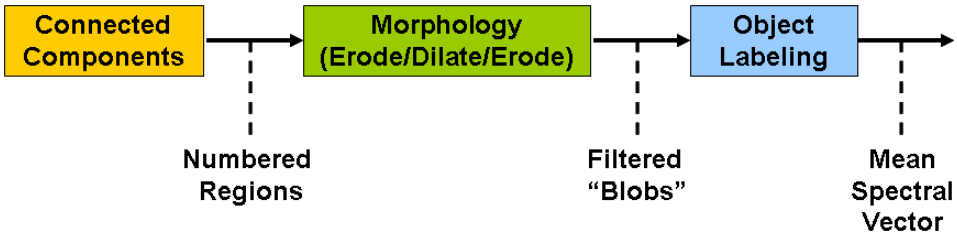


FIGURE 3.13 – Object Segmentation Flow Diagram.

An example frame is shown in figure 3.14, in which a pedestrian and vehicle have been detected as moving objects. The scene was captured from a single-band campus surveillance video and processed for motion detection using the spatiotemporal texture vectors technique described in the previous sections [Miezanko:2006].

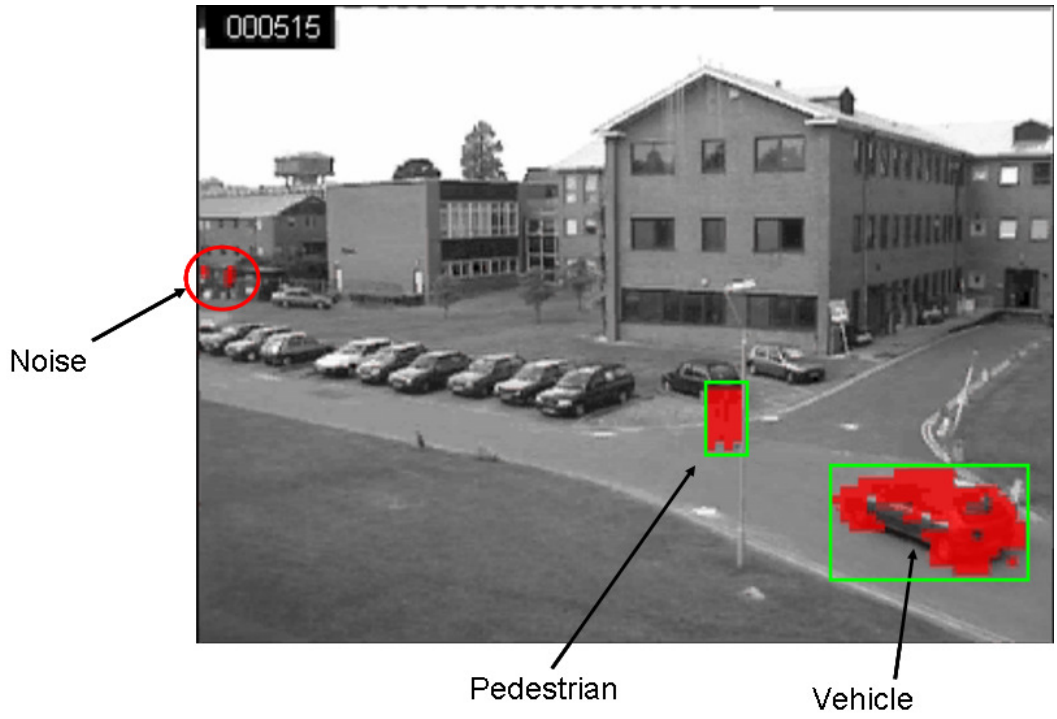


FIGURE 3.14 – Object Segmentation.

Notice the pixel regions to the left of the scene, which are considered noise and would be filtered out by this process. The two valid objects are then processed into blobs to ensure the entire object is represented. Although the entire vehicle was not detected as moving, morphological processing (not yet applied to this scene in this example) fills in the gaps and removes noisy pixels. Thus, the first step in the segmentation process is to process the moving pixels into these regions of connected components.

3.1.3.2 Connected Components Processing

The output from the motion detection process is in the form of a motion-matrix, as described earlier (figure 3.9). The motion-matrix is then converted to a sequence of logical image frames, with non-zero values where motion was detected. These individual values are processed using the connected components routine in Matlab (bwlabel), which groups neighboring pixels into regions; this process is applied to each frame in the sequence.

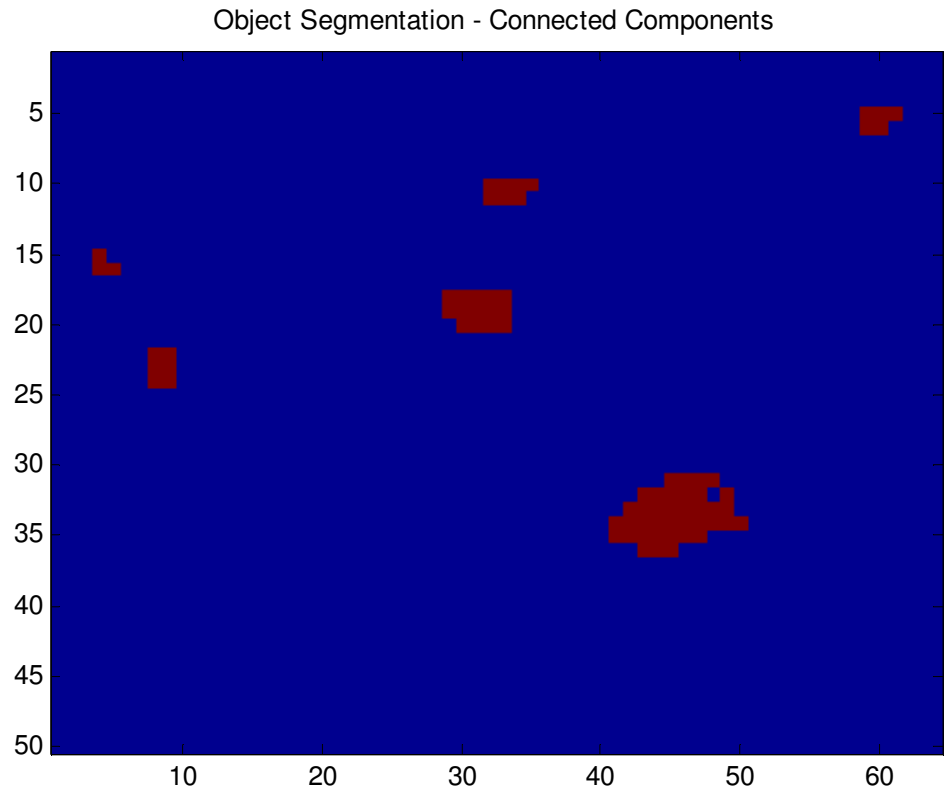


FIGURE 3.15 – Connected Components (WASPLITE Example Frame).

The output of this process is a sequence of frames that now have numbered regions, as shown in figure 3.15. Once these regions have been identified, each frame is filtered to refine them into numbered blobs.

3.1.3.3 Morphological Processing

The regions found in the connected components routine were found to be susceptible to noisy detections. Despite the fact that each frame was previously processed through a spectral filter, some anomalous objects were detected; these can be seen as small objects in figure 3.15. A visual inspection of the image frame revealed that some of these detections did not correlate to a valid target. Thus, morphological erosion followed by dilation removed the false detections, while filling in the valid objects. A final erosion step brought the dilated object back down to the correct size. A final connected components process produced filtered, numbered blobs, as seen in figure 3.16.

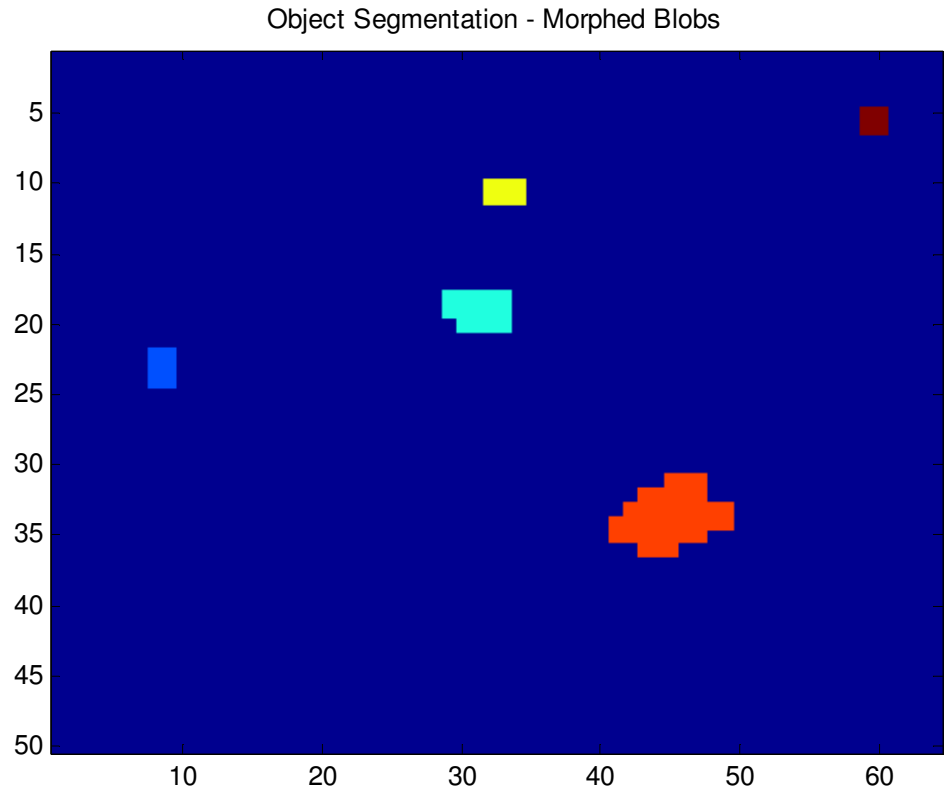


FIGURE 3.16 – Morphological Processing (WASPLITE Example Frame).

Notice that the initial connected components frame (figure 3.15) had six detected objects. However, after morphological processing there were only five (figure 3.16). Upon visual inspection of these frames, occasionally the removed object was actually a valid target (e.g. pedestrian or distant car). The settings for erosion and dilation were adjusted manually until a balance was achieved between removing noise and losing valid (albeit very small) targets. Generally, these valid targets would appear as a blob when they became large enough. Hence, after processing each frame into a valid set of numbered blobs, these objects could be identified and labeled for future use in object association.

3.1.3.4 Object Labeling

Once valid, numbered objects (or blobs) were established, bounding boxes were defined for each object. These boxes were based on the greatest extent of pixel locations in both the x- and y-direction. Given a bounding box, the centroid coordinates (x_c, y_c) defines the object location. Most importantly, the mean spectral vector for each object was saved for use in the object association step. Similar labels can be defined for other object attributes such as size (number of pixels), total number of objects in the frame, and object identification tags. The list of selected object attributes is shown in table 3.1.

Label #	Description	Notes
1	**Frame Number**	**Image File**
2	**Object Number**	**Matlab bwlabel**
3	**Centroid: x_c**	**Object Location**
4	**Centroid: y_c**	**Object Location**
5	**Number of Pixels**	**Object Size**
6	**Total Number of Objects**	**Per Frame**
7	***Mean Spectral Vector***	***Band 1 – Red***
8	***Mean Spectral Vector***	***Band 2 - Green***
9	***Mean Spectral Vector***	***Band 3 – Blue***
10	***Mean Spectral Vector***	***Band 4 – NIR***
11	***Mean Spectral Vector***	***Band 5- SWIR/LWIR***

TABLE 3.1 – List of Segmented Object Attributes.

The list of object attributes is by no means complete—further features could be defined and stored to assist object association, such as confidence measures based on spectral and spatial similarity (see future work section). Despite the expected increase in system performance in object detection and segmentation, it was assumed from the beginning of this project that object association would show the most compelling advantage when using multispectral data.

3.1.4 Object Association

Recall the underlying hypothesis of this project: Although system performance should decrease with frame rate, multispectral data should offset this disadvantage. This becomes especially true in the object association function. A complete object association function was not implemented in the end-to-end software for this project. However, it seems credible to assume that additional spectral information lends itself directly to better performance in distinguishing one detected object from another. Therefore, a statistical analysis of a subset of segmented object frames was used to verify the hypothesis of multispectral advantage. Accordingly, only the WASPLITE data was processed in order to validate the most challenging case.

The assessment of overall system performance was based on the theory that superior object association relates directly to tracking performance. To measure association (or matching) performance of objects in one frame to some future frame, a subset of detected object frames was chosen based on a truth assumption. The connected components routine used to identify and number the individual objects in a frame operates in the same manner every frame; the upper-left corner of the image is the first object and the lower-right hand corner is the last object.

The truth assumption is that the connect components routine identifies objects in the same order in each frame. Therefore, if the objects are being matched correctly, the first object in frame F1 should match (or overlap) the first object in frame F2. Even at very low frame rates, this assumption holds true for more than half the detected objects. The spatial distribution of moving objects can be expected to remain relatively stable over a time period on the order of one second. Additionally, only frames with three or more objects were chosen for processing. The truth assumption was verified visually; example frames are shown in figure 3.17.

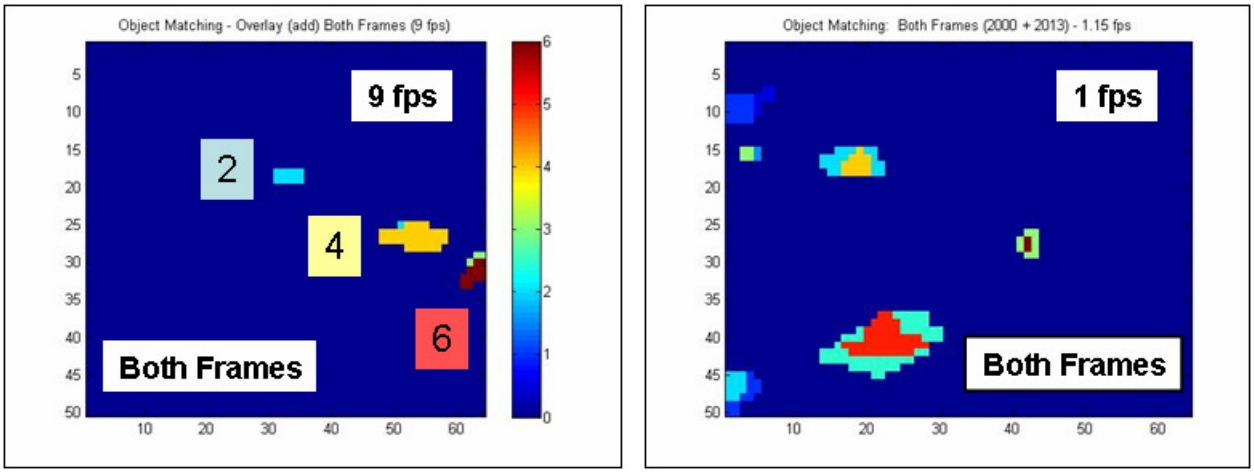


FIGURE 3.17 – Truth Assumption Verified by Overlapping Frames (F1 plus F2). Maximum Frame Rate (Left), Minimum Frame Rate (Right)

The case where the truth assumption is more often correct is when there is very little time between frames (i.e. maximum frame rate), as seen in figure 3.17 (left). In this example, three objects are obviously overlapping, and their combined object numbers correlate. However, even in the most difficult case (i.e. minimum frame rate), as shown in figure 3.17 (right), six objects were matched correctly.

The means of measuring matching performance required an approach different to simply counting correct and incorrect associations. In this case, a new variable, object separability (***ΔS***), is defined as the difference between the "best match" and the "next-best match" from one frame to the next, shown in equation 3.11. The first value in the equation is the magnitude of the spectral variation from object ***a*** in frame ***F1*** to object ***a*** in frame ***F2***. The second value is the spectral distance from object ***a*** in frame ***F1*** to the next-best match in frame ***F2*** (object ***b***).

$$\Delta S = (\ |F1a - F2a| \ - \ |F1a - F2b| \ ) \tag{3.11}$$

Euclidian distance was used for determining the magnitude of how "close" correctly matched objects were. In the single-band case, this distance is simply the one-dimensional difference; in the multispectral case, it is the root-mean-square (RMS) or Euclidian distance between two spectral vectors. Thus, ***ΔS*** accumulated for every object in every frame in the truth subset becomes the separability vector (SV). Thus, for each band/frame rate combination an SV was produced, which gives us 20 performance measures over five bands and four frame rate combinations.

To demonstrate separability between the two matching cases (best and next-best), figure 3.18 revisits one of the truth assumption examples. Here we find that object ***a*** matches well with object ***b*** in the next frame. However, object ***a*** matches best with object ***a*** in the next frame because it had the smallest spectral distance.

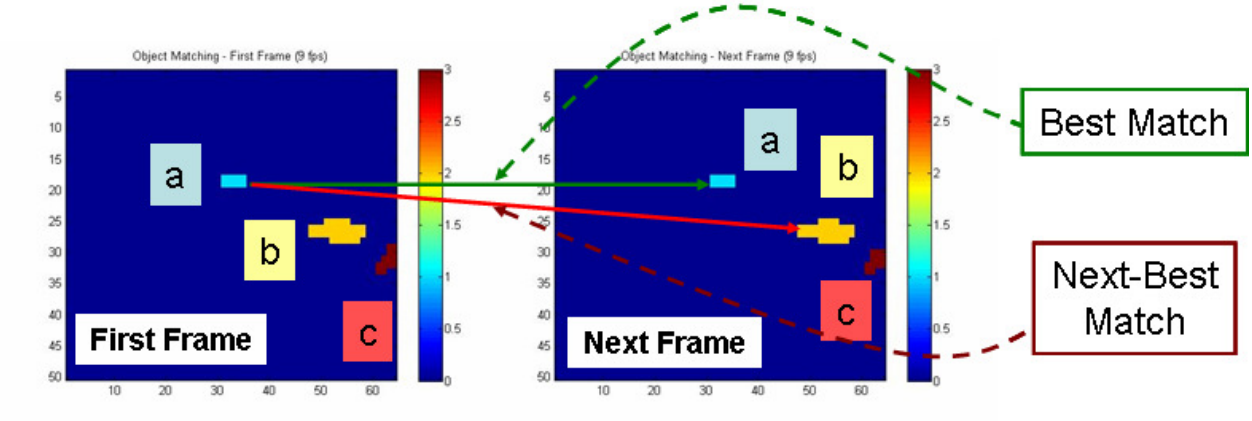


FIGURE 3.18 – Spectral Matching Example.

These comparisons were made at the maximum frame rate (9 fps) for every frame that met the truth assumption requirement. To assess lower frame rate performance, every second, fourth, and eighth frame was also matched to the initial frame. In this way, the same objects compared at 9 fps were being matched at lower frame rates (4.5, 2.25, and 1.13 fps, respectively). As stated earlier, improved system performance was expected when using spectral information for better object detection and segmentation. More importantly, the greatest improvement in system performance was expected in the object association subtask because it provides a unique spectral advantage.

3.1.5 Summary of System Model

The process described in this section breaks the three major surveillance system subtasks into their individual functions. A flow diagram in figure 3.19 provides an overview of this process. The first subtask—motion detection—formed the SP-vectors from the image data, reduced the dimensionality of these vectors, then used a dynamic threshold to find moving blocks of pixels.

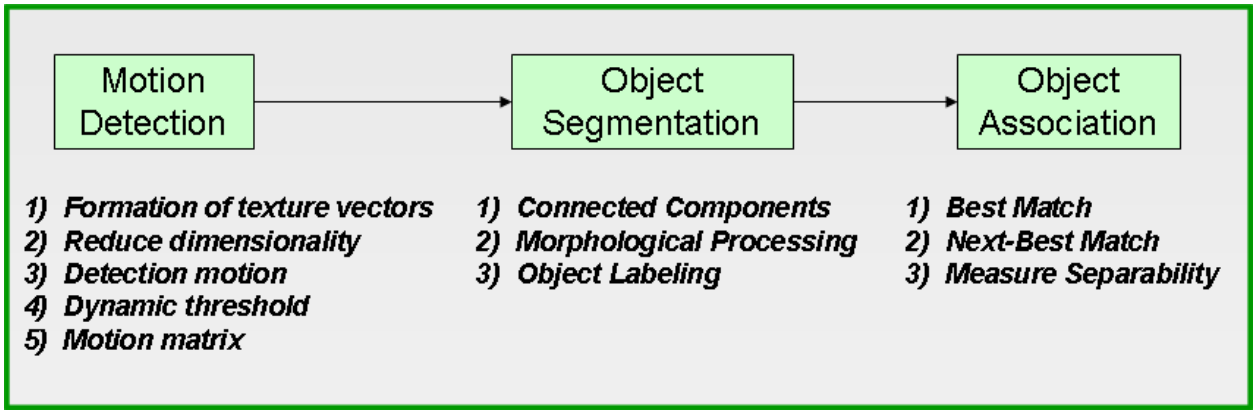


FIGURE 3.19 – System Subtasks Flow Diagram.

The second subtask—object segmentation—merged the detected pixels into separate "blobs", processed these blobs into distinct objects while removing noise, and labeled each object for later use. The third subtask—object association—compared the spectral distance from the best matching object to the next-best match in order to measure separability as a function of the number of bands. Note that the object association subtask was only applied to the real world WASPLITE data. Furthermore, the WASPLITE processing code was upgraded substantially from the DIRSIG process in order to overcome the additional noise due to sensor and registration issues. Once these three subtasks were implemented, a trade study of the results provided a means of comparing performance as a function of number of bands and frame rate.

3.2 Trade Study

The primary goal of the trade study was to evaluate surveillance system performance as a function of spectral and temporal resolution. It was expected that multispectral data would have an advantage over single-band sensors. A secondary goal was to validate the premise that equivalent (or better than) single-band performance might be achieved at a reduced frame rate by adding spectral detail to object appearance models. In this case, overall surveillance system performance was presumed to be directly related to moving object detection, segmentation, and object association performance. Recall from the introduction section that this is the stated hypothesis; here with the associated notional performance results we expected (figure 3.20).

It was desired that system results be presented as a performance surfaces, as seen in the notional surface plots in figure 3.20. Thus, system performance can be shown as a function of both number of bands and frame rate (fps) combinations. There are two ways to interpret system performance, the first being that general performance decreases with frame rate (x-axis) and increases with the number of bands (y-axis), as seen in figure 3.20 (top). Thus, better performance gets a higher score (i.e. up is good). Conversely, recalling the emphasis on reducing missed detections, figure 3.20 (bottom), shows the best performance at the lowest value (i.e. down is good).

Such a study would enable a surveillance system design to account for sensor specifications in the form of number of bands and desired frame rates required to achieve persistent surveillance of a given area of responsibility (AOR). Thus, the two main variables to consider in relation to system performance are: number of spectral bands (spectral resolution) and frame rate (temporal resolution). Other variables which might play an important role, such as spatial resolution, signal-to-noise (SNR), registration error, and computational complexity (runtimes), were not within the scope of this project (see future work section).

The reasoning behind the expected enhancement in performance is intuitive. Simply put, by increasing the number of spectral bands acquired, more information should lead to less uncertainty in moving object detection and association. Object detection becomes more sensitive given greater spatiotemporal detail in discerning the stationary background from the moving foreground. Greater sensitivity in detection becomes critical as the time between frames increases, allowing the detector to distinguish changes in background behavior over fewer observations. Similarly, object association—especially important with large time increments between frames—also improves, as demonstrated by the spectral separability measure. Higher confidence in spectral matching in addition to conventional spatial matching would inherently produce a better overall tracking system.

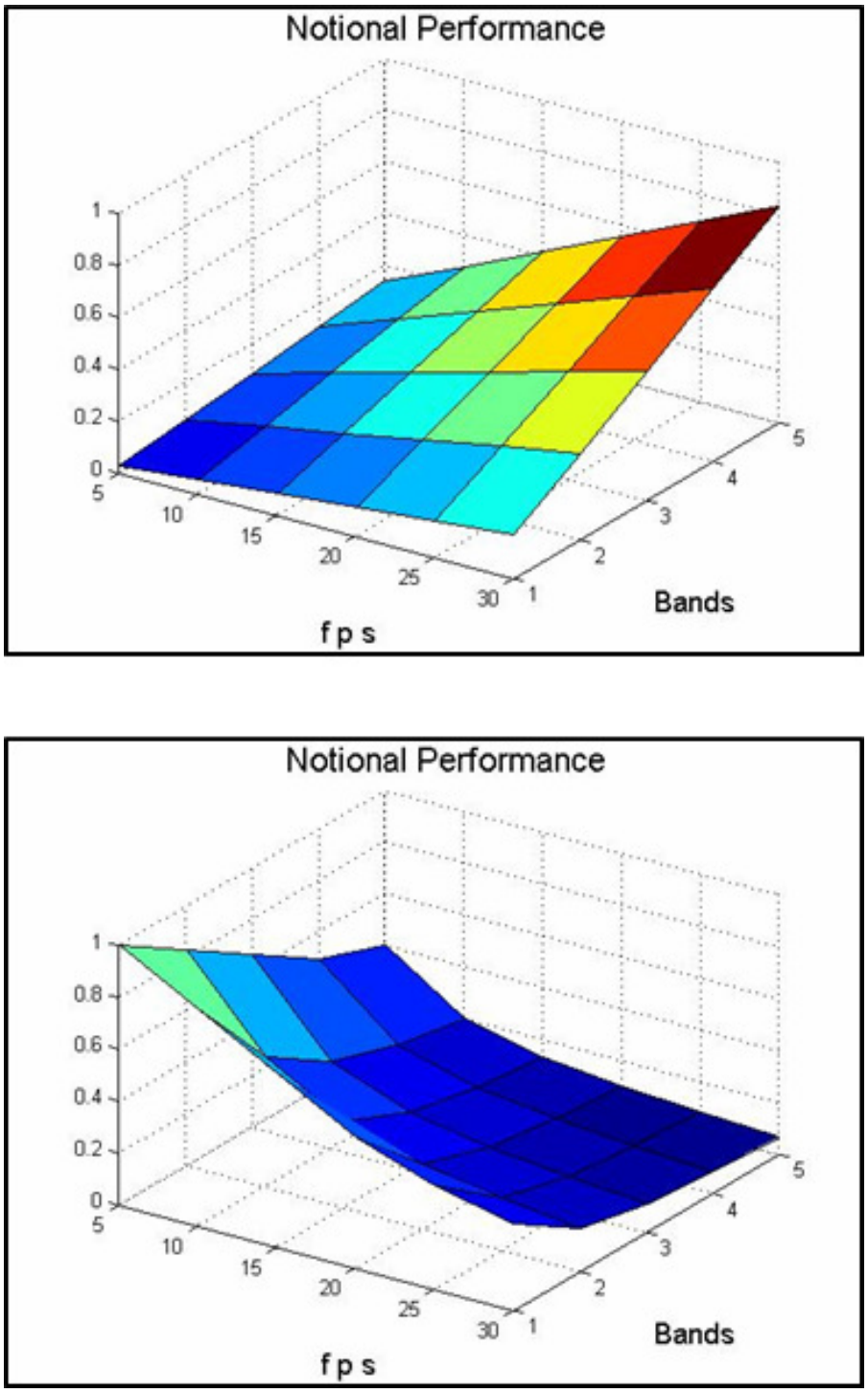


FIGURE 3.20 – Notional Results (System performance vs. number of bands, frame rate. Tracking Performance (Top), Missed Detections Performance (Bottom)

In contrast, reducing the frame rate—with more time elapsed between frames—increases the uncertainty in predicted object appearance, position, and velocity. Hence, system performance of a single-band system was expected to decrease. By adding spectral detail to offset the degradation due to reduced frame rate, the premise behind this trade study is sound. Thus, the starting point in this experiment was to adjust the spectral resolution variable.

3.2.1 Spectral Resolution

In order to assess the impact of the first variable (spectral resolution) on system performance, it was necessary to determine the limitations of a single-band system. Thus, the trade space was investigated using pan-chromatic visible band performance as a baseline. The single-band system was evaluated on the two datasets at the nominal video rate of 30 fps (DIRSIG) or 9 fps (WASPLITE). Then, system performance was evaluated by incrementally increasing spectral information. The additional bands were not selected in order of information content and/or contrast, because such an experiment was considered outside the scope of this project. Consequently, system performance was expressed as a function of the number of bands acquired. Additional spectral bands were added incrementally from the visible (Red, Green, Blue), to near infrared (NIR) and short-wave infrared (SWIR - DIRSIG) or long-wave infrared (LWIR - WASPLITE). The green-band was used as the single-band case simply because it is in the middle of the visible spectrum. The order of increasing spectral information is shown in table 3.2.

Number of Bands Processed	***Bands Used*** *(**R** = red, **G** = green, **B** = Blue, **NIR** = Near Infrared **SWIR** = Short-wave Infrared, **LWIR** = Thermal)*
1	***G***
2	***R, G***
3	***R, G, B***
4	***R, G, B, NIR***
5	***R, G, B, NIR, SWIR (DIRSIG)*** ***R, G, B, NIR, LWIR (WASPLITE)***

TABLE 3.2 – Order of Increasing Spectral Resolution.

Assuming that nominal single-band performance at maximum frame rate is already high, it was not assumed that including additional bands would improve performance significantly. However, the other thrust of the trade study was to compare system performance at reduced frame rates.

3.2.2 Temporal Resolution

Once baseline multispectral performance was established at maximum frame rate, the next set of iterations tested single-band performance by incrementally reducing frame rate until detection and tracking performance was noticeably affected. Finally, spectral bands were added one at a time, in the same order as before, and detection and tracking performance were evaluated again at incrementally reduced frame rates. Thus, system performance was first measured using two bands as a function of frame rate; then three bands, and so on until the system had been evaluated using all five bands.

The two datasets used had different maximum (default) frame rates; DIRSIG data was generated to represent 30 fps video, whereas WASPLITE data was limited to 9 fps due to system throughput. Reduced frame rates were achieved by simply skipping a variable number of frames between data points. For example, to get 1 fps for DIRSIG, every 30^{th} frame was considered; WASPLITE data was limited to taking every 9^{th} frame as a data point. Consequently, the DIRSIG data had a wider range of frame rates (down to a minimum of 1 fps) than the WASPLITE data. Table 3.3 summarizes the frame rates associated with the two datasets. Note that in order to get frame rates less than 1 fps (say 0.5 fps), the dataset would need more frames. In the case of this dataset, only 3,600 frames were available with truth data. To get 0.5 fps, every 18^{th} frame would be used, giving only 200 total frames of data—which was insufficient to run the motion detection algorithm.

System performance was expected to degrade as the frame rate decreased, as discussed in the background section. Therefore, system performance was expected to be highest at maximum frame rate (figure 3.22, top). However, tracking performance at low frame rates were also expected to recover as spectral bands were added. A higher priority was placed on reducing the number of missed objects at the expense of more false alarms (i.e. false alarms are more acceptable if missed objects are minimized). By emphasizing a low number of missed detections, system performance could also be measured by the minimum score (figure 3.22, bottom) – where a lower score means better performance.

The value of the trade study was revealed when system performance was evaluated in terms of both spectral and temporal resolution. Using simple, single valued-performance metrics, global comparisons of band/frame-rate combinations was made possible. Notional results, as discussed in the previous sections, were but a simplified prediction of system performance at the beginning of this project. However, the predicted surface plots are useful in discussing the actual experimental results later on.

Of course, system performance may also be affected by other variables such as: spatial resolution, noise (SNR), registration error, motion characteristics (partial/full occlusions, stop/go, mount/dismount, etc.), and scene complexity (spectral). Refer to the Future Work section for more on these topics.

Frame Stepping	DIRSIG = 30 fps / Step	WASPLITE = 9 fps / Step
Step 1 (every frame)	30	9
Step 2	15.0	4.5
Step 3	10.0	3
Step 4	7.5	2.25
Step 5	6.0	1.8
Step 6	5.0	1.5
Step 7	4.3	1.29
Step 8	3.75	1.12
Step 9	3.33	1.0
Step 10	3.0	(n/a)
Step 15	2.0	(n/a)
Step 26	1.15	(n/a)

TABLE 3.3 – Variable Frame Rates (DIRSIG vs. WASPLITE).

In summary, the main purpose of the trade study was to establish measures of system performance as a function of the number of bands and frame rate. System performance in the context of this study included motion detection, object segmentation, and object association. In order to test the methodology developed for this trade study, comprehensive, meaningful datasets were required. These datasets consisted of both real and simulated image sequences to emulate a multispectral surveillance system.

3.3 Datasets

Both synthetic and real world datasets were used to investigate the trade space. A synthetic scenario was developed with the intent of creating increasingly complex scenarios throughout the time sequence. In addition to perceptual complexity, as discussed in the previous section, spectral complexity was also addressed. The synthetic scene allows complete control of the object interactions, background appearance, and perfect knowledge of ground truth. Although real world datasets added the essential element of true spectral complexity, the realities of ground truth, image registration, and object interactions were fundamentally more challenging.

3.3.1 DIRSIG

Synthetic video sequences were developed for this project by using Digital Imaging and Remote Sensing Image Generation (DIRSIG) tools. Synthetic scenes generated by DIRSIG are based on a complex model which produces simulated images in the visible through thermal infrared regions. It can produce broad-band, multispectral and hyperspectral imagery through a set of radiation propagation subtasks. The DIRSIG simulation environment consists of exoatmospheric radiation sources, atmospheric and scene databases, and man-made sources, as seen in figure 3.21 [Ientilucci:2000] .

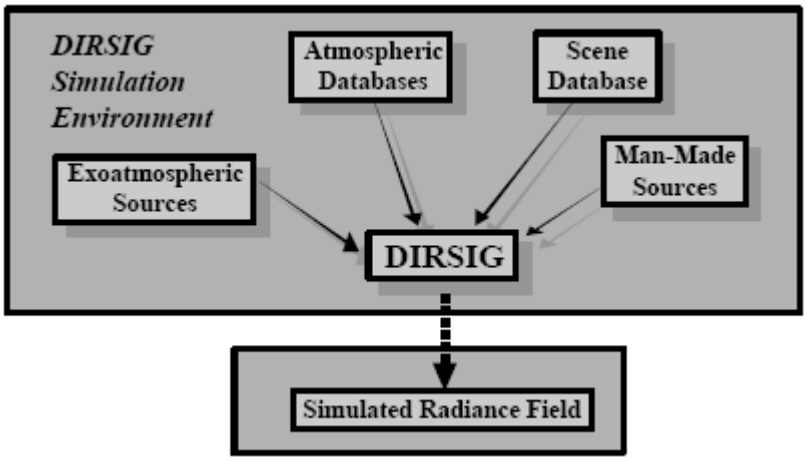


FIGURE 3.21 - Digital Imaging and Remote Sensing Image Generation (DIRSIG).

One purpose of this modeling effort is to generate imagery for testing spatial and spectral image exploitation algorithms. With the goal of reproducing imagery with sufficient spatial and spectral clutter comparable to real-world emissions, candidate algorithms can be extensively tested over a wide range of conditions at significantly lower cost compared to field collections. The second advantage of a synthetically generated dataset was the ability to produce perfect

ground truth—especially significant in testing algorithms for detecting and tracking the location of moving objects [DIRSIG:2006].

3.3.1.1 DIRSIG Movies

For the purposes of this research, the DIRSIG video sequence simulated five spectral bands: Red, Green, Blue, NIR, and SWIR. The sensor was specified to have 6" ground spatial distance (GSD) acquisition from a stationary, nadir looking platform. The synthetic video sequence takes into account spectral texture and clutter. The DIRSIG video was created without using ground truth modeling, which inherently has lower spatial and spectral complexity (or clutter) than a real world dataset would have.

FIGURE 3.22 – DIRSIG Video (1024 x1042) Image.

Scene content was managed in the form of spectral diversity of objects and in object interaction. Sufficient spectral diversity was desired in order to allow for object discrimination, including vehicles and pedestrians. Object movements and interactions were choreographed to simulate simple tracking tasks such as two vehicles passing each other in different directions to more complex tasks such as partial or total occlusions and passengers mounting and dismounting vehicles. The scenarios also evolved in object (or motion) clutter such that once an object enters the scene it remains for the duration of the video. In this way, the number of moving objects (both pedestrians and vehicles) continually increases, thus creating a continually more complex tracking environment.

A particular challenge in developing the synthetic environment was spectral clutter. The synthetic scene developed was not based on ground truth, thus the streets, sidewalks, grass, trees, and buildings, were all generic as can be seen in figure 3.22.

Given such a scene, the spectral clutter was not as complex as in real life. However, some manipulation was done to simulate spectral texture and spatial blurring. Given the 6" GSD requirement, the image size was established at (1024 x 1024) pixels in order to accommodate a sufficient number of moving objects. However, for the sake of processing efficiency, the frames were reduced to (256 x 256) pixels using bicubic interpolation (Matlab default). It is important to note here that when these frames are broken into (4 x 4) pixel blocks, the resulting frame resolution is (64 x 64) for the actual motion detection processing. An example of a single detection frame is shown in figure 3.23, where cars are easily recognized and pedestrians appear as one (or two) block targets.

Thus, the cars in the original scene are on the order of tens of pixels per target, which allows for a few blocks per target. However, the pedestrians (barely visible in the figure) are only a few pixels per target in the original frame, and become sub-pixel targets in the block image. Nevertheless, pedestrians are detected more often than not due to the sensitivity of the spatiotemporal detection method. Also note that the sun angle was set at midday to allow for some shadowing from trees and buildings, but did not pose a significant factor. Such shadowing, in addition to the occlusions made by trees, provided additional tracking challenges. Finally, there are several structures in the middle of the grassy areas—these are tunnels of varying lengths providing additional occlusions, which both partially and totally occlude vehicles as they drive through.

FIGURE 3.23 – DIRSIG Scene Reduced to (64 x 64) Block Images.

The DIRSIG movie developed for this project was constructed from 4,400 frames with duration of about 2.4 minutes at 30 fps. The original ".png" image frames were converted to ".tif" files in order to be read into Matlab. There are two types of moving objects in the scene: Cars and pedestrians, with cars being more spectrally diverse than the people. Despite the attention given to spatial and spectral diversity, preliminary motion detection results indicated that the synthetic data was too perfect. Thus, random noise was added to the synthetic data to provide additional variability resembling real sensor noise.

3.3.1.2 DIRSIG Signal to Noise

Initial results using the DIRSIG synthetic movie frames indicated that the data might be too perfect. As an experimental adjustment to the data, random Gaussian noise was added to each frame. Signal to Noise (SNR) of 250 was used in order to adjust the pixel values (0 – 255) by ± 1 brightness values. The results did not change drastically, but additional detections were discovered. The difference in the motion-matrix results between the noisy data and the noise free data is shown in figure 3.24, where the detections shown are not seen in the noiseless data. These additional detections are apparently the result of noise, seen as additional variability in the motion detection algorithm.

Given that noisy frames produce additional (possibly false) detections adds legitimacy to the prospect that adding noise better represents real world data. Thus, having generated a suitably realistic dataset, the next task was to develop a truth dataset with which to compare detection results.

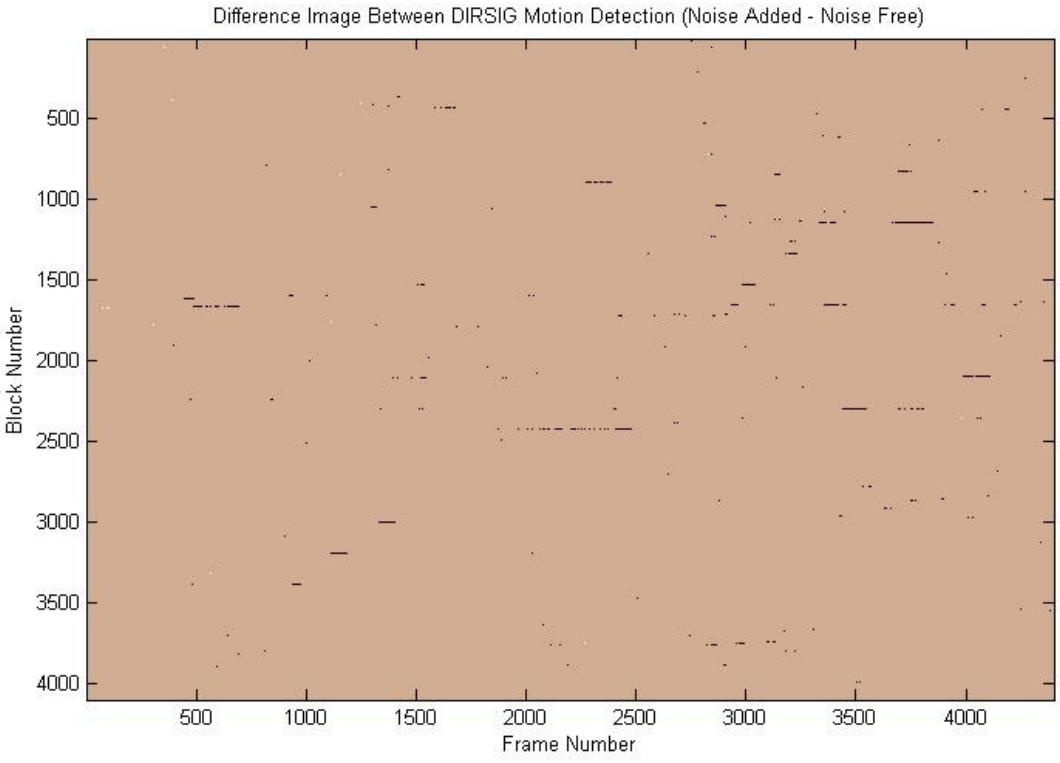


FIGURE 3.24 – Motion Detection Difference Image: Noisy vs. Noiseless Data.

3.3.1.3 DIRSIG Motion Truth

One main advantage of using synthetic data is that there is perfect motion truth available. In the case of the DIRSIG movie, each frame was composed of the same background pixels with moving objects placed over the scene at the appropriate location for each time increment. Despite the noise added to the scene, the background image was very consistent over the entire video sequence. Thus, simple frame differencing to remove the background produced truth objects for each frame in the video sequence, as seen in figure 3.25.

FIGURE 3.25 – DIRSIG Motion Truth Frame (1024 x 1024).

Because the motion truth frames were constructed directly from the source data, the images were the same (1024 x 1024) resolution as the original dataset. Likewise, these truth frames were resized to correlate to the (64 x 64) block-space images. Again, bicubic interpolation was used and verified as the more desirable result (figure 3.26). The images in this figure are zoomed images of the subspace outlined (in orange) in the previous figure.

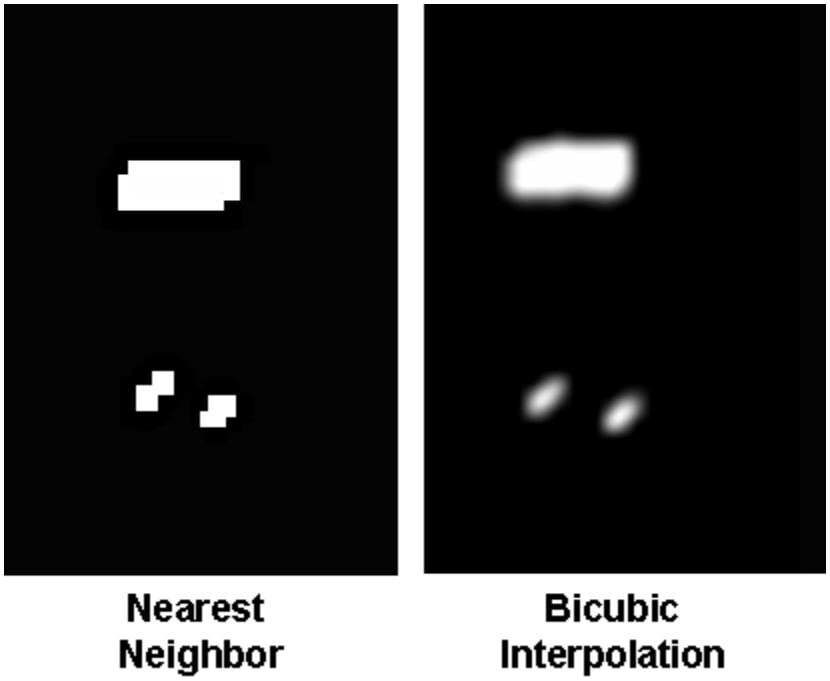


FIGURE 3.26 – Motion Truth Resized to (64 x 64) Block-Space (Zoom of Figure 3.25).

By observation of the two resized images, the nearest neighbor averaging provided truth objects that were angular and box-like (figure 3.26, left). However, the bicubic interpolation provided objects that were more rounded, which included a little more of the surrounding information (figure 3. 26, right). The bicubic result provides more flexibility in thresholding exactly where the object outline should be defined. Thus, bicubic interpolation was applied to reduce the spatial resolution of truth images, and an appropriate threshold was determined to produce "blobs" rather than boxes for object outlines.

Having developed a process to detect and segment moving objects using DIRSIG multispectral data, the results were compared to the motion truth data, as will be discussed in the results section. The final step in evaluating overall system performance in the trade study was to apply this same process in the real world.

3.3.2 Real World Data

In order to capture similar scenarios as generated by DIRSIG, a stable, stationary "airborne" platform was required. Similar to the synthetic scenes, a variety of moving objects was desired. To accomplish this, a multi-band, high frame rate sensor was placed on top of a building looking down on a parking lot with both pedestrians and vehicles. To exploit the real-world scenario, collections were planned during periods of the day that would provide an appropriate amount of activity. Thus, the data collection was timed for periods of both low and high activity. Another advantage of using a real sensor was that we could collect much longer datasets than the few

minutes of DIRSIG video (which took several weeks to generate). The primary disadvantages of multispectral sensor data were less than perfect image registration and limitations on frame rates. Finding a suitable imaging sensor became the next challenge.

3.3.2.1 WASP & WASPLITE Overview

The Center for Imaging Science (CIS) has access to two imaging sensors, developed locally by the Laboratory for Imaging Algorithms and Systems (LIAS). The LIAS group's primary focus is the research and implementation of data processing algorithms as well as the systems which encompass those algorithms. Of the two LIAS imaging systems available, one was selected as the most appropriate for collecting data for this trade study.

3.3.2.1.1 WASP

The Wildfire Airborne Sensor Program (WASP) is a sensor platform originally intended to identify wildfires by collecting imagery in the visible through thermal infrared wavelengths. Ground sample distance (GSD) is about six inches (visible) at a flying height of 10,000 feet. GSD in the infrared is degraded to about 48 inches. Figure 3.27 shows the underside of the WASP system as installed in an aircraft.

FIGURE 3.27 – WASP Sensor System.

The WASP system combines four infrared and high-resolution mapping cameras that sweep across the line of flight, taking a series of individual images. Each camera images a different spectral band: three infrared cameras in the short-wave, mid-wave and long-wave IR, and a high-resolution digital camera maps the terrain in the visible spectrum. The image data is corrected, registered, and ortho-rectified to ground coordinates in real time [WASP:2006]. Although the

WASP system was not used to collect data for this project, it is the predecessor of the more portable system selected.

3.3.2.1.2 WASPLITE

The WASPLITE system was designed as a portable, more specialized version of the WASP system—providing greater flexibility in set-up, configuration, and other user specific requirements (figure 3.28). The design principles were that it was to be smaller and less expensive than the WASP system, and at the same time provide greater flexibility in other applications. Although this smaller sensor can be flown in a variety of small aircraft, it operates in a fixed configuration rather than using a gimbal [WASPLITE:2007]. However, the portability and variety of spectral configurations made this instrument an ideal choice to simulate a stationary airborne platform.

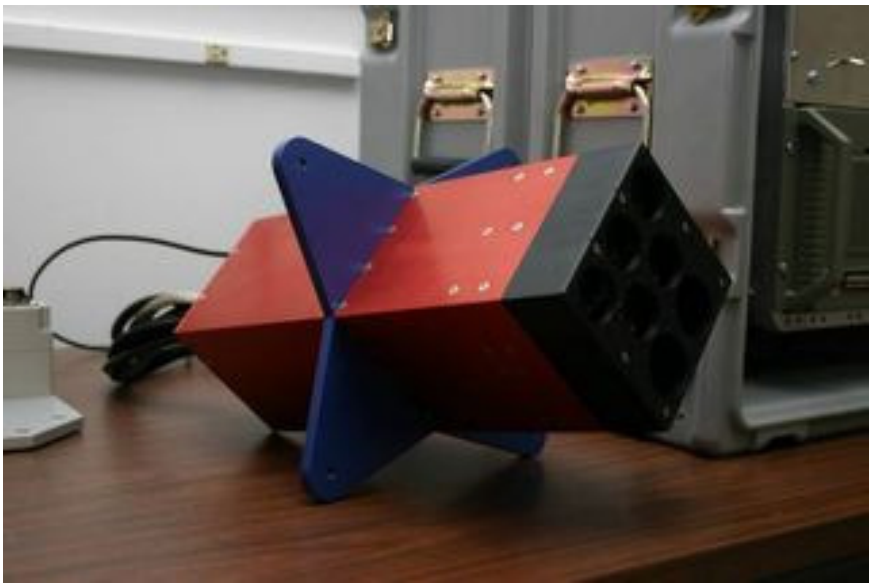


FIGURE 3.28 – WASPLITE Imaging System.

The WASPLITE sensor is actually composed of seven separate imaging systems, five of which are unfiltered panchromatic imagers (see figure 3.29). These five cameras are capable of having a separate spectral filter attached in front of each lens. In addition to the visible spectrum cameras, there are two infrared sensors: A short-wave infrared (SWIR) camera and a long-wave (LWIR) thermal bolometer.

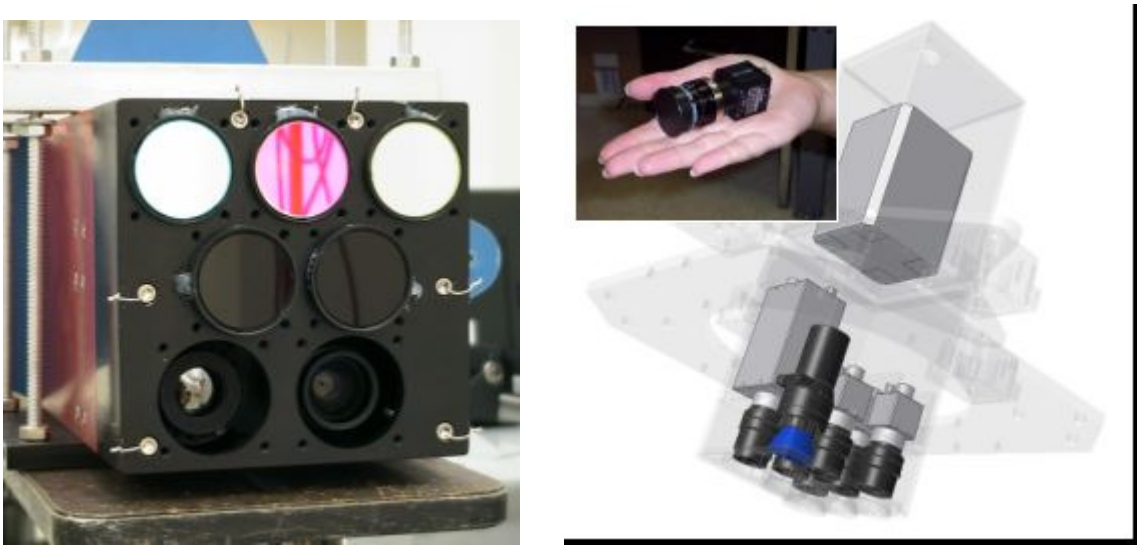


FIGURE 3.29 – WASPLITE Sensors.

For the purposes of this experiment, four of the cameras were filtered to provide red, green, blue, and near infrared (NIR) images. Finally, the thermal bolometer provided the fifth and final band in the data collection. Table 3.4 summarizes these sensors and camera response measurements ***[Bartlett:2006].*** It is interesting to note that calibration results of the cameras showed all three visible band filters had some residual sensitivity at 800-1000 nm, which overlaps the NIR band.

Dichroic-Filter	Band-pass
Red	**600 – 800 nm**
Green	**500 – 600 nm**
Blue	**400 – 500 nm**
NIR	**695 – 950 nm**
LWIR	**8 – 12 μm**

TABLE 3.4 – WASPLITE Camera Filters.

Two system performance parameters were constrained by the WASPLITE system: Spatial and temporal resolution. The cameras provided approximately (640 x 480) pixel images, in contrast to the (1024 x 1024) DIRSIG images. Furthermore, the LWIR images were only (320 x 240) pixels. Thus, the fifth band in this dataset had only one fourth the spatial resolution of the other four bands. Additionally, the data throughput of the instrument was essentially limited by the hard-drive capability to store the data. As such, the maximum frame rate achievable was approximately nine frames per second. In contrast to DIRSIG data, the real world WASPLITE data had lower spatial and temporal resolution. Spectral resolution, on the other hand, was

equivalent because both DIRSIG and WASPLITE used a simple, broad average over each spectral channel. The WASPLITE system was made available for this project, which enabled collection, registration, and processing of real world data.

3.3.2.2 WASPLITE Data Collection

As described in the previous section, the WASPLITE system provided video data in five bands (R, G, B, NIR, and LWIR) with a maximum frame rate of approximately nine frames per second (9 fps). The datasets were collected from the top of Building 76 on the RIT campus looking down on a parking lot with both pedestrians and vehicles. The WASPLITE instrument configuration and rooftop setup can be seen in figure 3.30.

FIGURE 3.30 – Instrument Setup (WASPLITE Data Collection).

In order to get a variety of activity in the scene, two collections were planned to be executed at midday on 23 July 2007. An example data frame is shown in figure 3.31. Unlike the DIRSIG data, the size of moving objects varies throughout the scene. Notice that closer objects (cars at bottom of the image) are significantly bigger than distant objects (such as vehicles parked further away or traveling along the roadway near the top of the image frame).

FIGURE 3.31 – Example Frame (WASPLITE Collection).

The first collection was taken just prior to noon, when the parking lots and sidewalks were basically clear of people and traffic. This provided a relatively low amount of motion activity. The second collection was conducted just after noon to observe as motion activity increased with lunch hour traffic. Similar to the DIRSIG data, a noontime sun angle provided a minimal amount of object shadowing on the ground. Also similar to the DIRSIG dataset, the second WASPLITE collection provided a video sequence increasing in spatial clutter as more moving objects entered the scene over time. Thus, the second data collection was deemed superior to the first and was processed for this project.

The WASPLITE dataset chosen for this project consisted of 9,200 frames capturing about 17 minutes of video at 9 fps. Although the five spectral band image streams were collected simultaneously, they were generated from five separate sensors. Fortunately, the problem of registering WASPLITE data was solved by a former CIS student [McNamara:2007]. The resulting IDL code was used for registering the data from this collection. Upon visual inspection of the overlaid frames, the quality of registration for the purposes of this experiment was in question. The transformed images appeared to be accurate only to within a few pixels of each other, as seen in three (RGB) of the five bands in figure 3.32 (left). In the zoom image (right), the pedestrians are well defined. However, the colors composing the sign-post (circled in yellow) are offset vertically, with red to the left and green to the right of the post. Thus, WASPLITE registration appears to be accurate to within two or three pixels.

FIGURE 3.32 – WASPLITE Image Registration (RGB; 3-Band Example).

However, initial motion detection results demonstrated that registration error did not pose a significant factor. There are a few reasons for this apparent insensitivity to registration error. First, the image registration seemed better at the center of the images, and got worse toward the edges. This problem was overcome by cropping the images from (640 x 480) pixels down to (512 x 400) pixels. Second, because the images were then resized to (256 x 200) for processing efficiency, some of the misalignment was absorbed in the bicubic interpolation. Third, the resized images are then processed in (4 x 4) pixel blocks, which further average the results into (64 x 50) block images. Given a suitable dataset, the next step in processing was to establish motion truth; similar in function to the DIRSIG data. However, defining motion truth for the WASPLITE data proved to be more challenging because it lacked perfect knowledge of the background.

3.3.2.3 WASPLITE Motion Truth

In order to establish motion truth for the WASPLITE data, a simple background subtraction—as used for DIRSIG data—would not suffice. In addition to forming a less-than-perfect background model, morphology and manual processing were also required.

To get a reasonable background model in the context of post-processing the entire dataset, it was possible to "look into the future" and use proceeding frames to estimate the current background. For each set of 100 frames processed by background subtraction, the background model was formed by using the median image of those same 100 frames—very similar to the spectral filtering process. However, in the motion truth case, future frames were used instead of previous frames. An example of a WASPLITE data frame can be seen in figure 3.33 (top). Notice the areas with moving objects (circled in yellow) have been removed in the background model frame (figure 3.33, bottom).

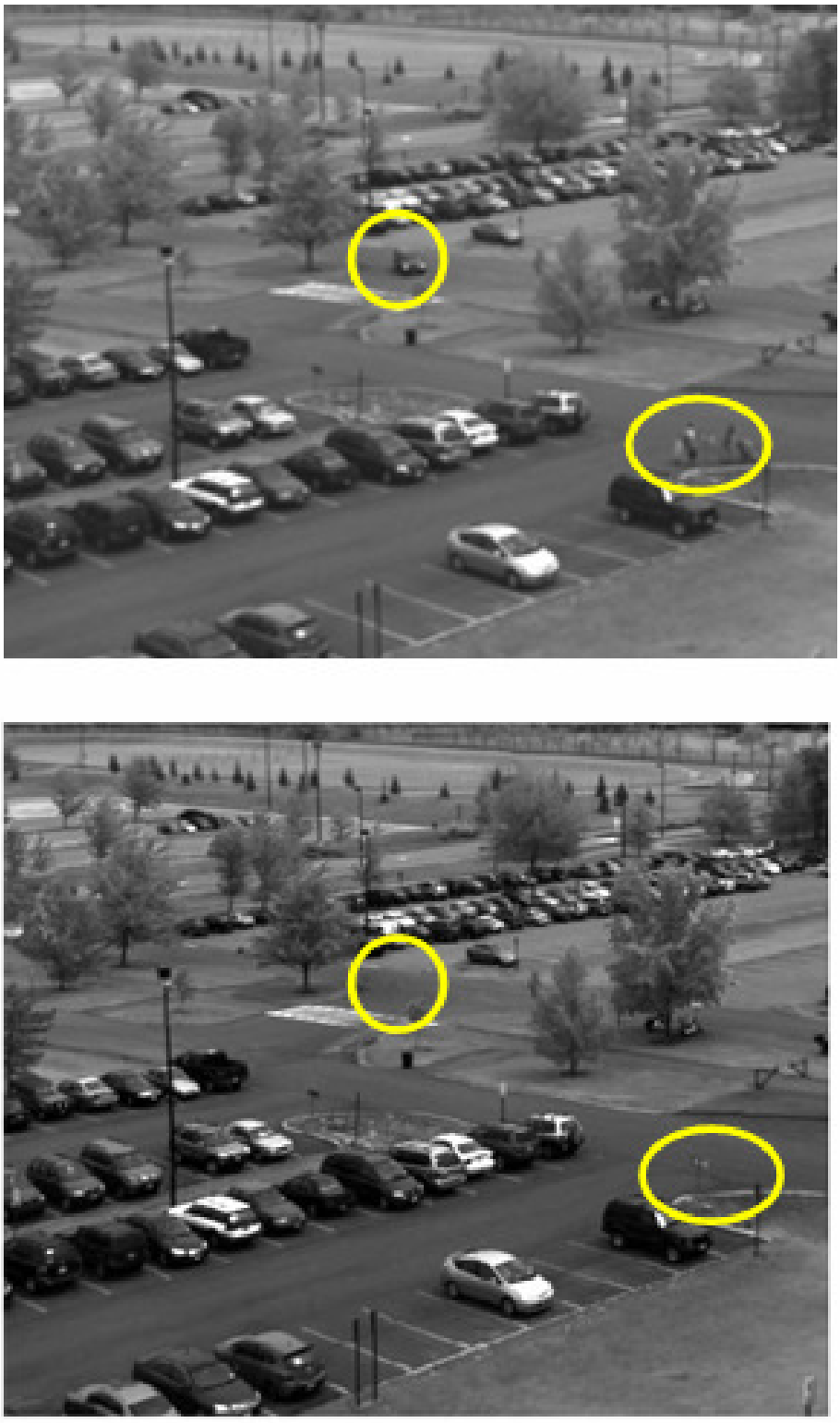


FIGURE 3.33 – Original Image (Top) vs. Background Model (Bottom).

Morphological processing was then implemented to remove noisy pixels in the motion truth frame (similar to object segmentation of the actual detection results). Here, the same noise found in detected motion needed to be removed from the motion truth data. By eroding, dilating, and eroding the objects, speckle noise was removed and the authentic moving objects were filled in and reduced to correct size. Figure 3.34 shows the motion truth frame resulting from background subtraction followed by morphological processing. Notice in particular that it not only shows the two moving regions found earlier by visual observation (circled in yellow), but two other moving objects are also revealed (boxed in red).

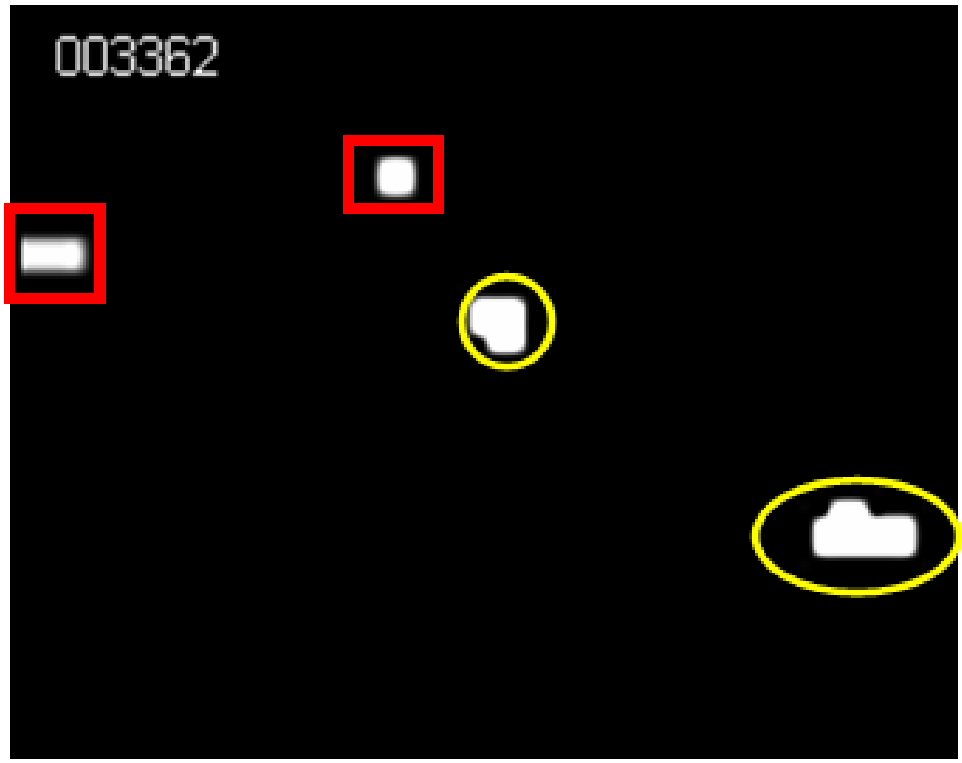


FIGURE 3.34 – Motion Truth after Processing.

Although background subtraction and morphology removed most of the noise and highlighted moving objects, certain scenarios caused object "shadows" to remain for no more than 100 frames at a time. If a car was parked and began to move during this interval, the parked car became part of the background model, and the ground surface pixels appeared as a new object when the car moved away. Conversely, if a moving car stopped during this interval, the parked car continued to be different from the background, despite the fact that it was no longer moving.

In this case, the last resort was to manually fill-in the incorrectly marked objects. An image editing program was used to observe each frame and remove these shadow objects and any noise that was not filtered out by morphology. Lacking a more efficient method to "cheat" and get

good motion truth, only about 3,700 truth frames were manually processed. However, the number of frames was equivalent to the DIRSIG dataset and deemed acceptable for the purposes of this project.

The two datasets utilized for this trade study provided the required spectral contrast and temporal variability to asses the trade space. Once the motion truth frames were generated for both WASPLITE and DIRSIG datasets, moving object detection, segmentation, and association performance could be evaluated. However, performance metrics posed another challenge in assembling the results in a meaningful way.

3.4 Performance Metrics

Suitable performance metrics were required for a successful investigation of the trade space being considered. An essential element to evaluating moving object detection and segmentation performance was motion truth, as described in the previous section. As stated at the beginning of this project, the goal of this research was to present a single-valued "score" for each band/frame rate combination. Doing so provides us with a means to compare system performance as a function of both variables. With such a metric (or set of metrics) we can produce performance surfaces for each surveillance subtask: Motion detection, object segmentation, and object association.

Each function can be considered a filtering process. First, raw input data was formed into a logical mask of moving pixels (or blocks of pixels), discarding the background pixels. Second, the tagged moving pixels were assembled and shaped into moving objects and labeled; isolated detections were filtered out as noise (or in some cases as very small objects). Third, the spectral mean of selected objects was compared to the spectra of potential matches in future frames, providing an overall sense of how well a tracking system would perform given multispectral data.

3.4.1 Motion Detection Metrics

At the motion detection level, (4 x 4) blocks of pixels were processed and tagged as either moving or stationary. In order to assess the accuracy of these detections, the motion tags were compared directly to motion truth. Figure 3.35 shows an example run of WASPLITE motion-matrices at maximum frame rate (9 fps). The assessment was accomplished by directly comparing the truth motion-matrix (figure 3.35, left) to the detected motion-matrix (figure 3.35, right).

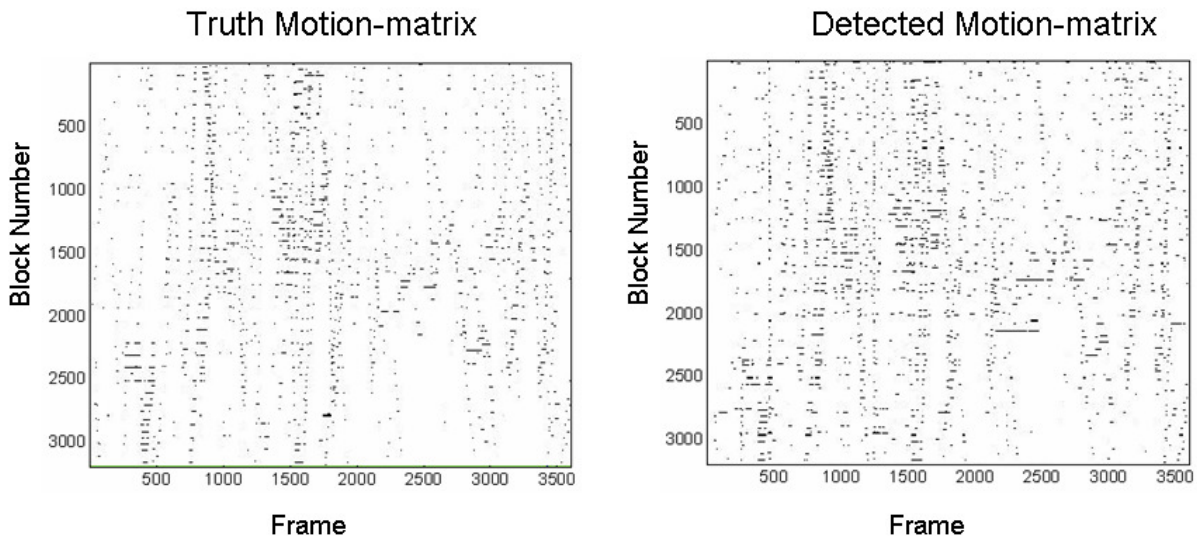


FIGURE 3.35 – Motion-matrix Comparison (Left - Truth, Right - Detected).

Both of these logical matrices were converted to signed integers (int8) in Matlab. Then the detected motion-matrix was subtracted from the truth motion-matrix. An example of the resulting difference matrix is shown in figure 3.36.

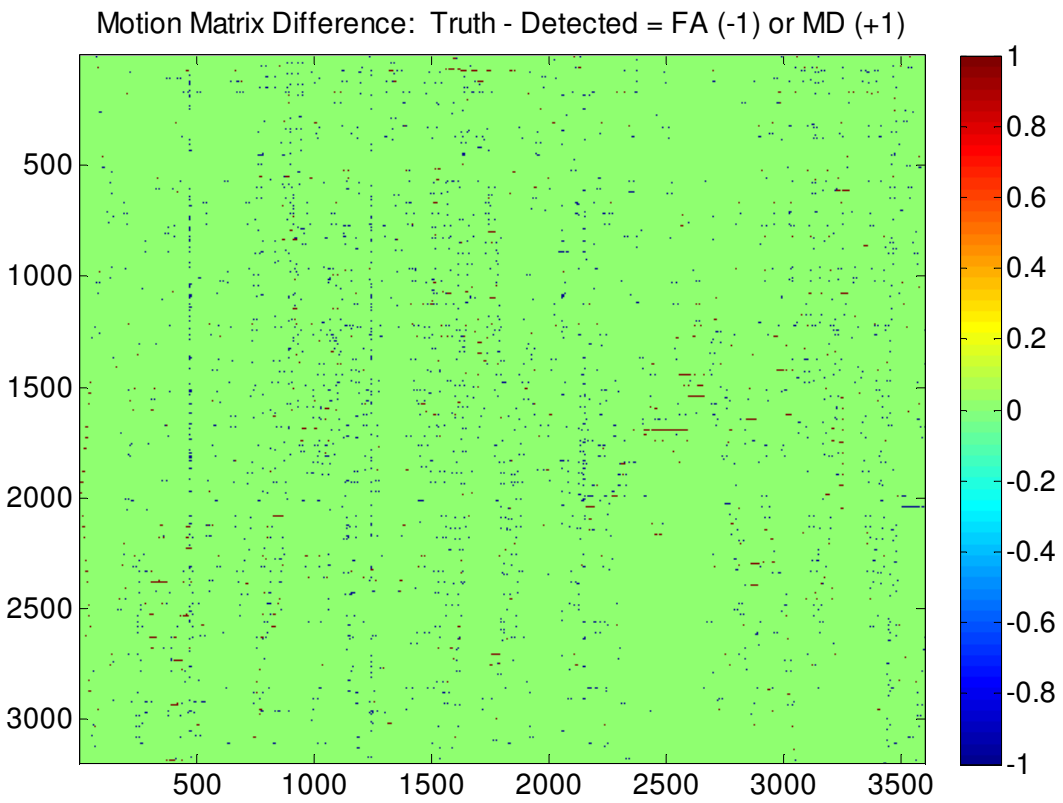


FIGURE 3.36 – Motion-matrix Difference (Truth – Detected).

In doing so, any resulting zeros must be either background pixels (0 – 0 = 0) or valid motion (1 – 1 = 0). More importantly, negative values represent false alarms because a detection was found where no motion truth was present (0 – 1 = -1). Conversely, positive values in the difference matrix were missed detections because motion truth occurred where no detections were found (1 – 0 = 1). It is valuable to reassert here that all of the processing settings were based on reducing missed detection. As can be seen in figure 3.36, the majority of incorrect results were false alarms (blue = -1), with very few missed detections (red = +1).

To better understand this metric, it was helpful to plot the number of false alarms (FA), missed detections (MD), and actual truth "hits" (MT) as a function of frame number. Keep in mind this is the single result of processing all frames in a video sequence, so these results represent one band/frame rate combination. (The example results presented above were at 30 fps for the 2-band case). The three variables (FA, MD, and MT) were plotted on the same graph to show relative scale in numbers of occurrences per frame, as seen in figure 3.37.

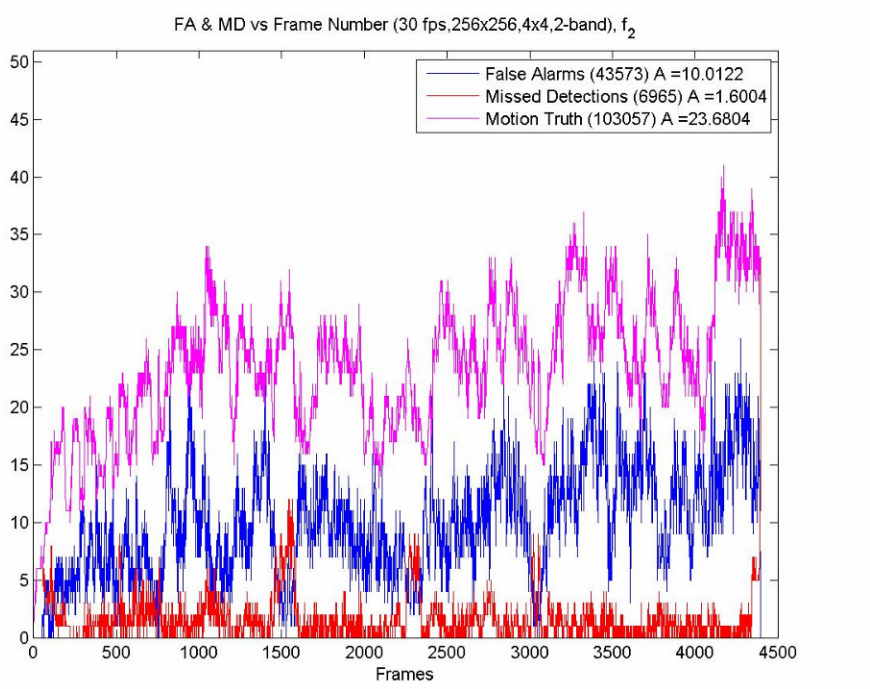


FIGURE 3.37 – False Alarms, Missed Detections, and Motion Truth vs. Frame.

Although this plot gives us a sense of relative values, it does not provide the single-valued "score" desired for this particular example (30 fps, 2-band). However, by taking the area under each curve, a single value for each variable is produced. We can then compare the FA and MD values for each band/frame rate combination (whereas motion truth remains constant for this dataset). The final data product was a surface plot for FA and MD, respectively, as a function of number of bands and frame rate (see example, figure 3.38).

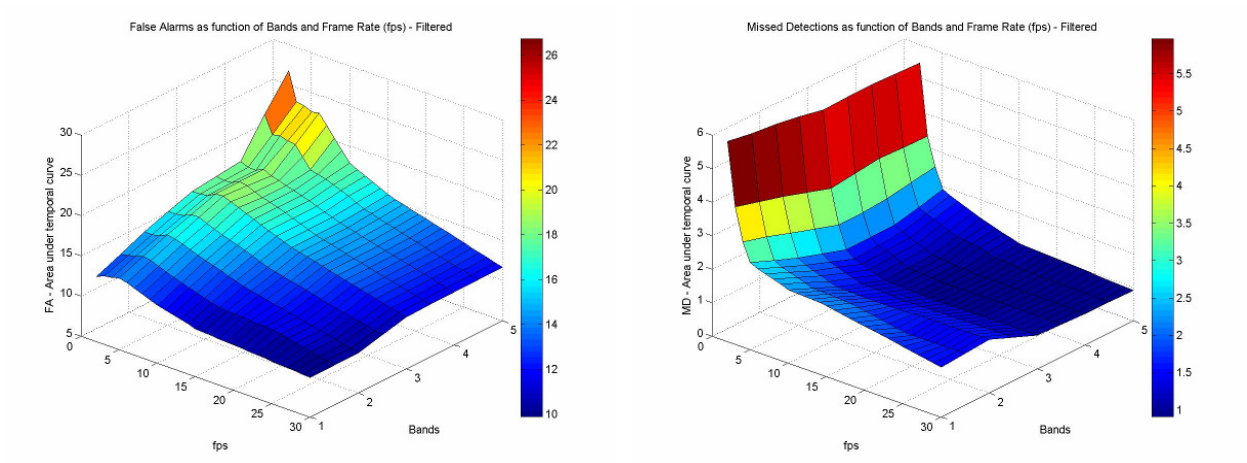


FIGURE 3.38 – Performance Surfaces: False Alarms (Left), Missed Detections (Right).

Given a suitable means of measuring motion detection performance, object segmentation performance was the next variable to measure. An evaluation was required on how well detected pixels were segmented into objects. Similar to motion detection, the object segmentation was compared to the truth data and evaluated over all bands and frame rates.

3.4.2 Object Segmentation Metrics

Having measured motion detection performance at the pixel level, object level results were produced in the same fashion. In this case, performance surfaces were produced in the form of false objects (FO) and missed objects (MO). Motion truth "blobs" were compared to the detected and segmented blobs by finding the intersection of both image frames. Each frame in the video sequence had a blob-image as a result of the segmentation process. As logical images, truth and detected blobs were added together producing a value of two (1 + 1 = 2) wherever there was object overlap (figure 3.39).

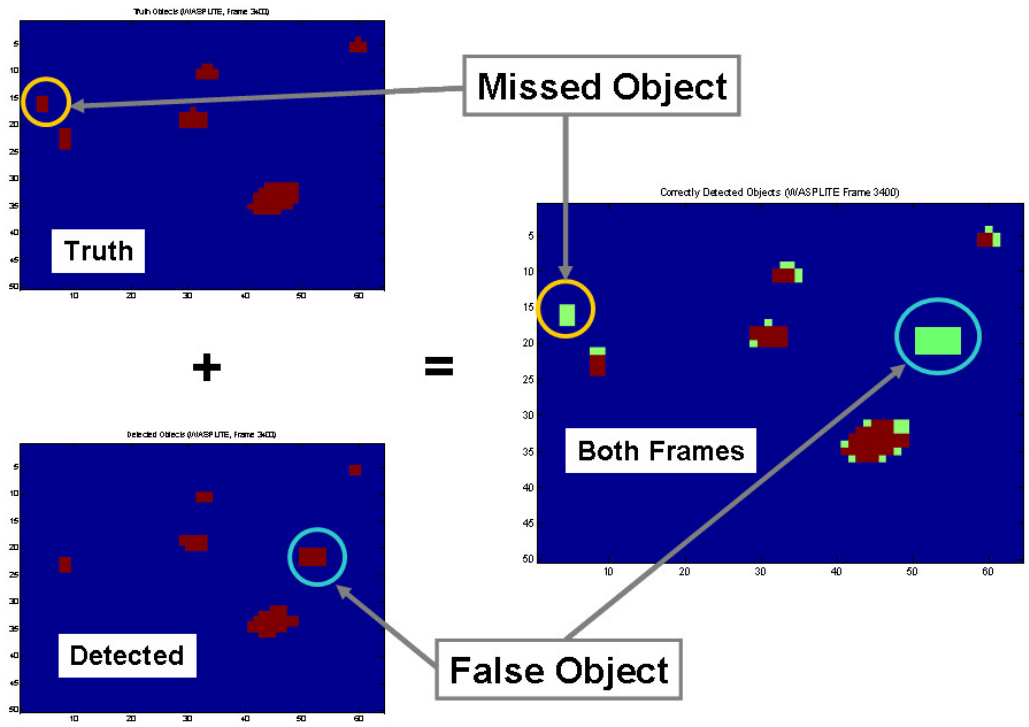


FIGURE 3.39 – Overlap of Truth and Detected Objects (WASPLITE Example).

The number of objects found in this combined image represented the number of "correct" hits for that frame. It follows logically that the number of missed objects (MO) was equal to the number of truth objects not found (i.e. truth – correct = missed). Conversely, the number of false

objects (FO) was equal to the number of detected objects that didn't agree with truth (i.e. detected – correct = false); see Matalab code in figure 3.40.

Matlab Code----------

```
Correct_hits(c) = num_c;
Data_hits(c)    = num_d;
Truth_hits(c)   = num_t;

mo(c) = num_t - num_c;
fo(c) = num_d - num_c;

mo_n = sum(mo)/Total_truth;
fo_n = sum(fo)/Total_truth;
```

FIGURE 3.40 – Matlab Code for Object Segmentation Metrics.

Because the FO and MO count for each frame rate was an accumulation over the total number of frames, low frame rate results had a lower total number of objects. To accommodate this, these counts were normalized by the total number of objects processed in that particular video sequence. Thus, we have an object level single value score for FO and MO, respectively, similar to FA and MD. Performance surfaces were produced in exactly the same fashion as for the pixel level results (figure 3.41).

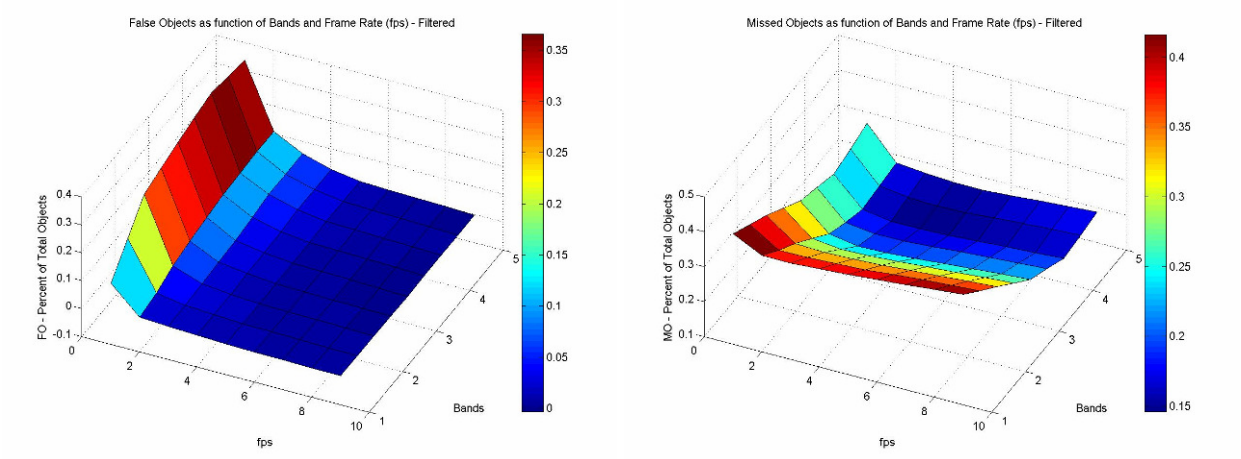


FIGURE 3.41 – Performance Surfaces: False Objects (Left), Missed Objects (Right).

Given object level performance surfaces—which correlate directly to the pixel level surfaces—we now have both global and frame-scale perspectives on system performance thus far. The final subtask, which was expected to produce the greatest spectral advantage, is object association.

3.4.3 Object Association Metrics

As described in the methodology section, a separability vector (SV) was accumulated for each band/frame rate combination (20 in all). The SV represents a single separability score for every object that was successfully matched to an object in a future frame. To get an overall separability "score" for each band/frame rate combination, histograms were made of each separability vector (SV), as seen in figure 3.42. The first intuition was to take the mean value of each separability vector, as seen in the vertical line on each example. For the one-band case (9fps), the mean separability was 14.74 (left), whereas the five-band case had a separability of 39.15 (right). Given a single value for each band/frame rate combination, a performance surface was generated as a function of number of bands and frame rate, similar to the detection and segmentation results.

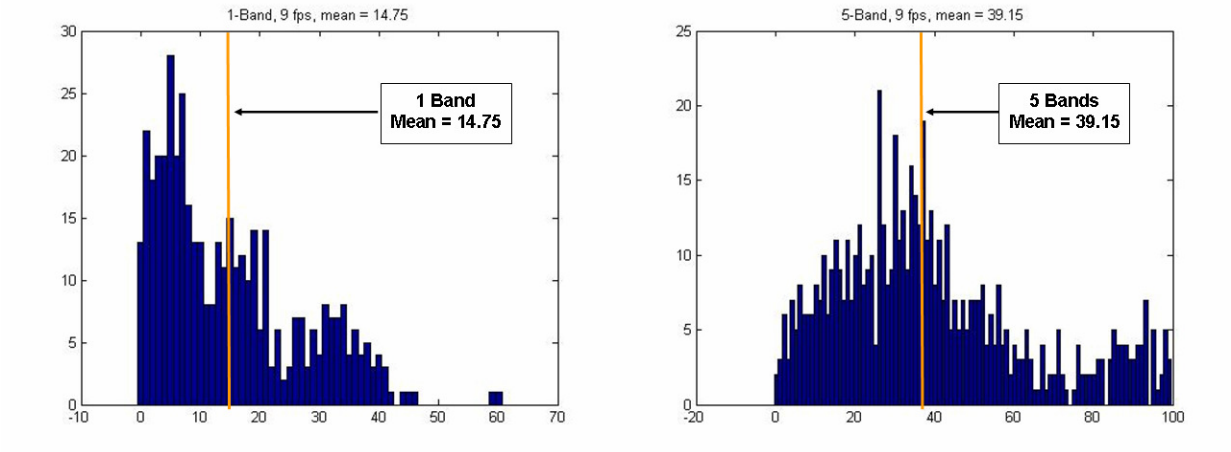


FIGURE 3.42 – Histograms of Separability Vector (WASPLITE Example).

However, the above histograms have more information than the simple mean value. Although the mean SV score relates directly to the ability of distinguishing between two objects, these values are not necessarily in the same scale. It seems these scores should be normalized to better compare them. As a first effort to get the SV scores into the same space, they were normalized by the standard deviation of each relative histogram. In this way, the histograms are rescaled to have unit variance. Figure 3.43 shows the same two example histograms in the new standard deviation measurement space; note the x-axis is now in numbers of standard deviation. In this case, the difference between mean SV values is less dramatic. In these revised histograms, a normal density curve was added to give a further impression of each distribution (although the data is not necessarily Gaussian)

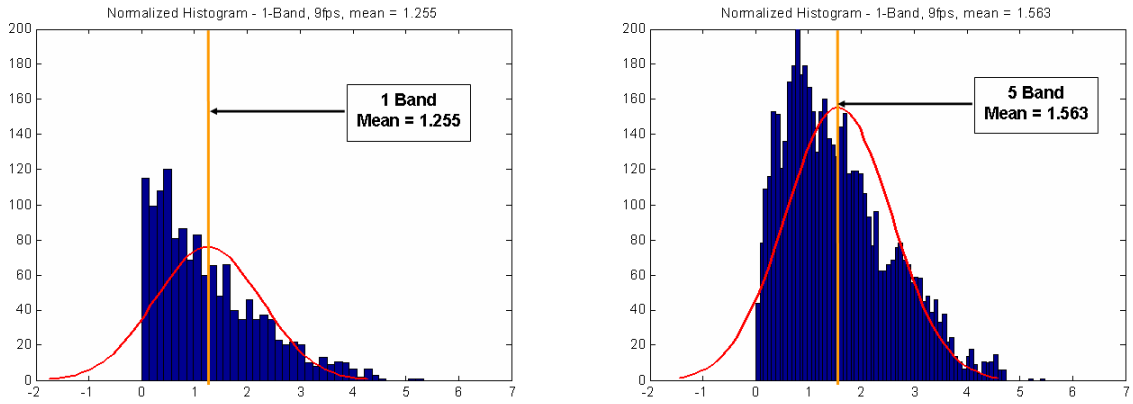


FIGURE 3.43 – Normalized Histograms of Separability Vector (WASPLITE Example).

It is important to make two comments regarding this normalization. First, the normalization essentially puts each histogram (and mean value) into a "sigma-space" (i.e. relative to each distribution). However, notice that the distribution for the multiple bands not only tends to move right (good), it also tends to be more spread out. At first, normalization seems to put the SV scores on a fair playing field. Yet, the higher dimensional cases actually detected more objects, thus had more targets to compare. Observe in figure 3.43 that the normalized histograms are presented with the same x- and y- dimensions. However, the five-band histogram (right) has far more counts than the one-band case (left). In other words, the spectral advantage in detecting and segmenting objects led to a wider distribution in the SV histograms with a greater number of matched objects. This is because multiple-band cases are actually comparing new and different targets in addition to the single-band targets. Thus, normalizing by the standard deviation may not be the correct choice, but it sufficed as a first attempt at comparing the SV scores in a different way.

Second, the higher frame rate case (9 fps) had more frames thus more objects to compare. Consequently, the low frame rate worst case (1.13 fps) had far less objects to compare. It would be fair to say that this worst case might exhibit sampling error in that an insufficient number of samples were available to get a statistically sound result.

In summary, performance surfaces were generated for each surveillance system subtask: Motion detection; Object segmentation; and Object Association. These metrics were applied to both the synthetic (DIRSIG) data and the real world collection (WASPLITE). A final caveat on the results as they are presented: The original detection and segmentation routines were applied to the DIRSIG data. When the WASPLITE data became available, these two subtasks were revised and improved based on the DIRSIG results. Although the general methodology was maintained, it became evident that the detection and segmentation settings were data dependent. The experience gained through processing the DIRSIG data provided lessons learned in the WASPLITE process. Variable settings such as the number of principle components to keep, dynamic threshold settings, and motion truth thresholds were all adjusted accordingly. Finally, preprocessing of the raw data was streamlined to reduce redundant calculations. Additionally,

the object association subtask was only applied to the WASPLITE data, because initial results using the synthetic data indicated that it did not have sufficient spectral variability to make the object matching meaningful.

Having found sufficient means to measure system performance at the subtask level (motion detection, object segmentation, and object association), the trade study was made possible. A comparison of the subtask results and motion truth provided an assessment of performance relative to the number of bands and frame rate used. The results of applying and improving the subtask algorithms to the test data will be evaluated in the next section.

Chapter 4

Results

4.1 Results Overview

Having developed a methodology to measure surveillance system performance and assembled the required datasets, the results of the trade study can be broken into four parts. First, the spectral filter addressed the issue of invalid detections due to the "ghosting" effect at low frame rates. Once this issue was solved, the spectral filter was applied to all data prior to the three main subtasks of the system. Second, motion detection performance measured pixel-level results in the context of missed detections (MD) versus false alarms (FA). Third, object segmentation performance extended the pixel-level results to object-level results in the form of missed objects (MO) versus false objects (FO). Fourth, an object association comparison was made using a subset of the entire WASPLITE dataset which conformed to certain truth assumptions. The subtasks can be envisioned as a set of filters for the input data, as seen in figure 4.1, whereby each stage in the process minimized missed detections at the expense of slightly higher false alarms.

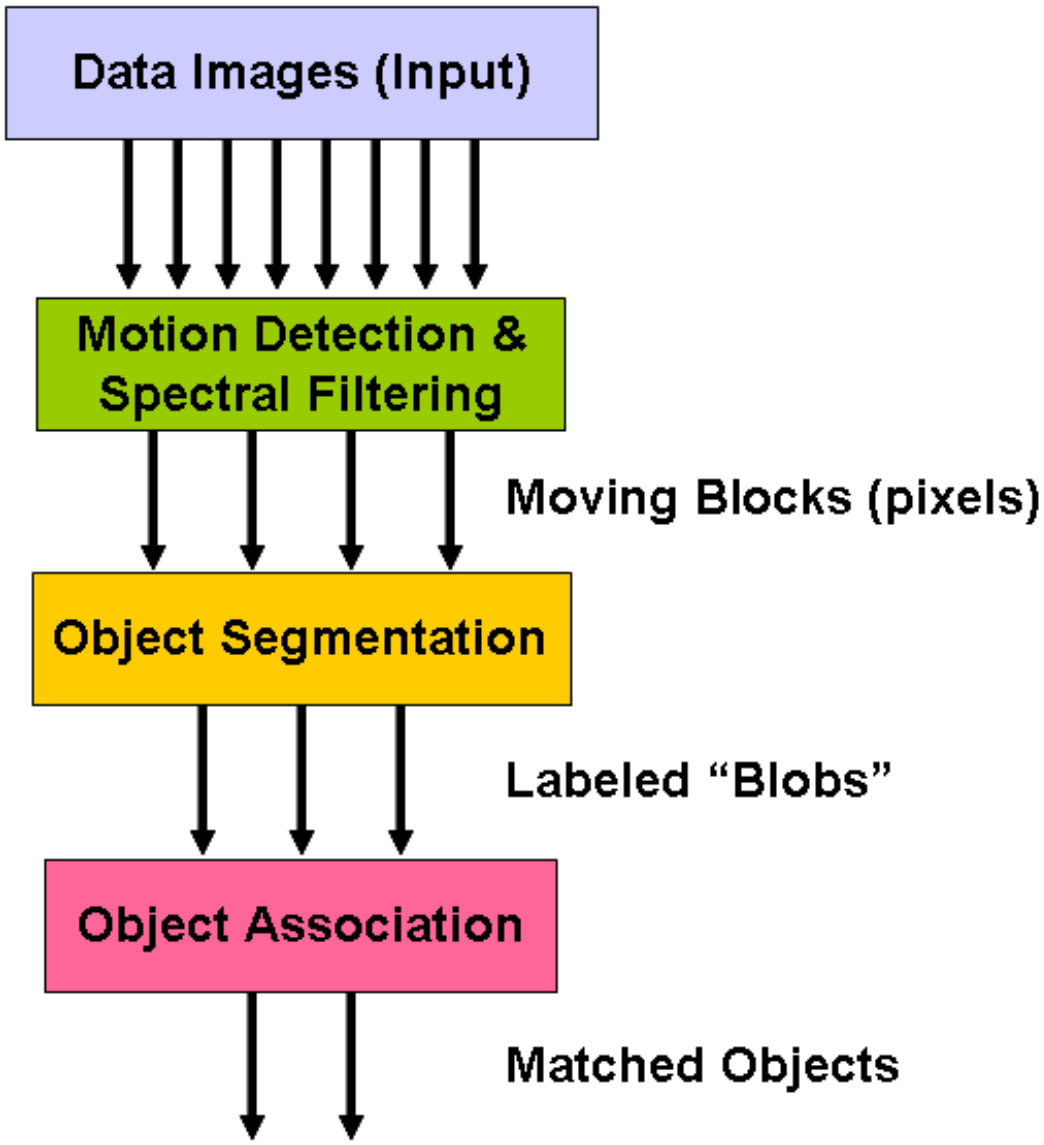


FIGURE 4.1 – System Flow Diagram (Input Data Filter).

Raw image files compose the input video sequence, which are first processed to tag moving blocks (4 x 4 spatial windows of pixels). Once stationary blocks have been eliminated, a spectral filter is applied to remove ghost detections (considered a sub-function of the detection process). The next step is to combine neighboring blocks into segmented objects, filtering out noisy detections. The final subtask evaluates a subset of the detected objects in order to measure the spectral advantage in object association.

Note that the first two subtasks (detection and segmentation) were conducted using both DIRSIG and WASPLITE datasets. However, the object association evaluation was only performed on the WASPLITE data because the synthetic data did not provide enough spectral variability to make separability measurements meaningful. As such, the results are presented in order of DIRSIG results followed by a similar set of results (plus object association) for the WASPLITE data.

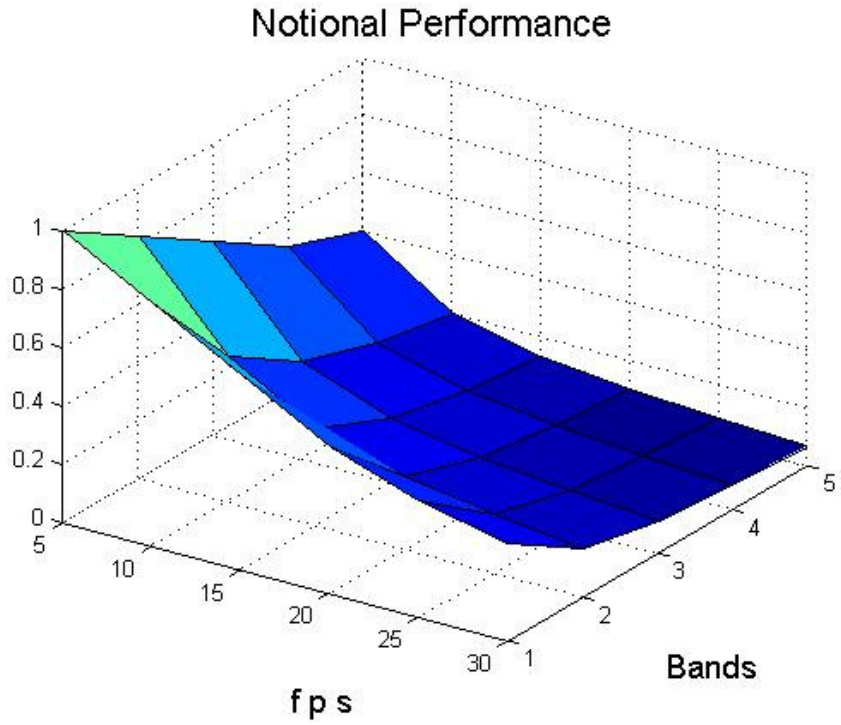


FIGURE 4.2 – Notional Missed Detection Results.

It is valuable to revisit the underlying hypothesis and notional results before reviewing the actual performance results. The basic premise is clear: More bands (to a point) should provide better system performance. In the context of reducing the number of missed detections, better performance is seen as a lower score (figure 4.2). The corollary to this hypothesis is that the spectral advantage would offset performance degradation at low frame rates.

4.2 Spectral Filter Results

The first challenge in low frame rate motion detection and tracking was encountered in the initial results. A phenomenon referred to as "ghosting" manifested in low frame rate results due to a feature of the spatiotemporal texture vectors technique. Because the technique evaluates a temporal window of seven frames, an inherent lag in processing includes detections both fore and aft of the current spatiotemporal location. Although it is present in high frame rate data, it is not as apparent when large objects are moving relatively slowly. Conversely, at low frame rates, objects are "moving" quickly and demonstrate the ghost-detections more dramatically.

4.2.1 Multispectral Data at Maximum Frame Rate

The first examination of the DIRSIG results at maximum frame rate (30 fps) showed little improvement by using more bands. The result was not unexpected in that the single-band spatiotemporal vector works very well at standard video frame rate. In fact, as a result of the literature review, this particular algorithm was selected because it was found to out-perform more popular algorithms [Latecki_Miezianko:2006].

Preliminary results using DIRSIG, however, opened the question of whether or not keeping the top ten principle components (i.e. the largest ten eigenvalues) was sufficient for reducing the dimensionality of multispectral input. The synthetic data indicated that keeping the top ten principle components (PCs) was appropriate for the data content. An analysis of the accumulated eigenvalues in the single-band case showed that keeping ten PCs captured roughly 99.1 % of the information content (figure 4.3, left). The same analysis on the five-band case showed that about 98.9% of the information was still retained (figure 4.3, right).

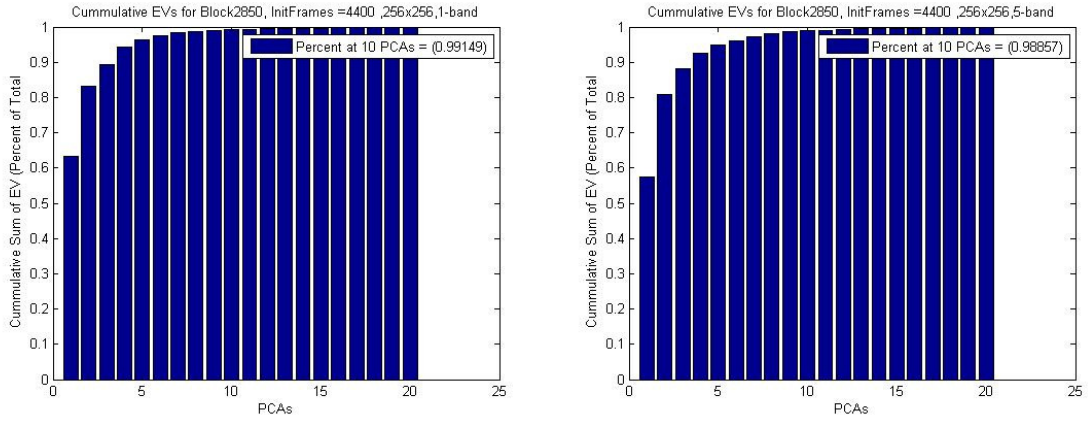


FIGURE 4.3 – DIRSIG Principle Component Analysis. Single-band (Left) and Five-band (Right) Using DIRSIG data

In the extreme case for the DIRSIG dataset, the multi-band cumulative value reached 99.9% of the information content by keeping up to 24 PCs. However, further studies showed little difference in motion detection performance when using as many as 20 PCs compared to the baseline of 10 PCs. Additionally, computing the larger covariance matrices resulted in much longer runtimes. Therefore, the baseline detector was set to keep only the ten largest eigenvalues when computing the PCA transformation matrices for DIRSIG data, which maintained the same 99% information content used in the original single-band algorithm. The 99% information content threshold was used to evaluate the multispectral cases.

However, when the same PCA comparison was applied to WASPLITE data, there was a definite dependence of information content captured in the eigenvalues. Even the single-band case needed more than 10 PCs to retain 99% of the information content. Essentially, each band setting required additional PCs in proportion to the number of bands being processed. With more spectral variability in the real world data, preserving 99% of the information content required 14 PCs for the single-band case (figure 4.4, left) and 25 PCs for the five-band case (figure 4.4, right). The two-, three-, and four-band cases required 18, 20, and 22 PCs, respectively, to maintain the 99% information threshold.

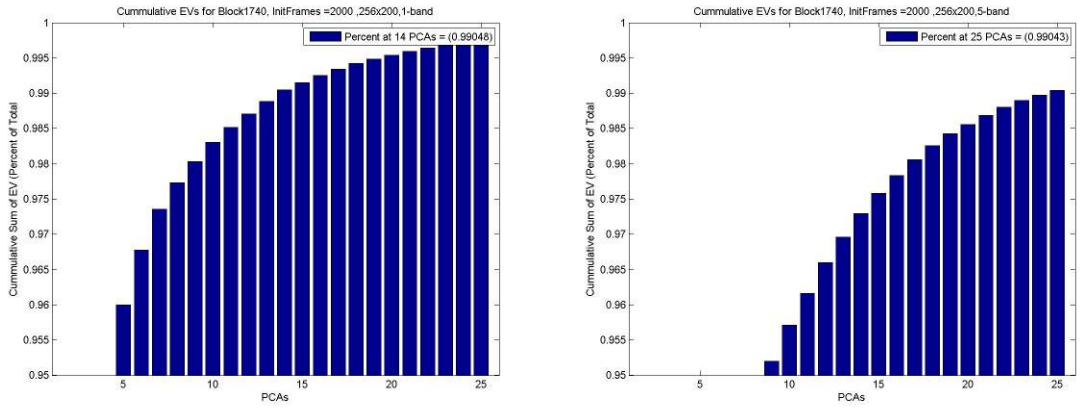


FIGURE 4.4 – WASPLITE Principle Component Analysis. Single-band (Left) and Five-band (Right) Using DIRSIG data

It is interesting to note that with real data, the spectral content is decidedly more variable than the synthetic data. The cumulative eigenvalue curves demonstrate this even more dramatically when comparing the one-band case to the five-band case. The curve for the multispectral data (figure 4.4, right) is much shallower, reaching the 99% threshold well after the single-band curve (figure 4.4, left). This makes sense because principle components represent the direction of maximum variability within a multi-dimensional vector space. With more spectral variability in the real data, the requirement for more PCs follows with additional variability spreading into higher principle components. With spatiotemporal texture vectors (SP-vectors) reduced appropriately, motion was detected based on the temporal variability of these new vectors.

4.2.2 Single Band Detection at Low Frame Rate

When motion detection results were first evaluated at low frame rates, a phenomenon called "ghosting" was observed (as described in the methodology section). The reason for this was intimated earlier that the seven-frame temporal window was to blame. When a large and/or slow object enters the seven frame temporal window, it causes a change in variability sufficient to be identified as motion (figure 4.5). In this case, the moving object occupies the temporal window for many frames, producing a motion measure, $mm(t_i)$ at that spatial location for that particular frame (f_i).

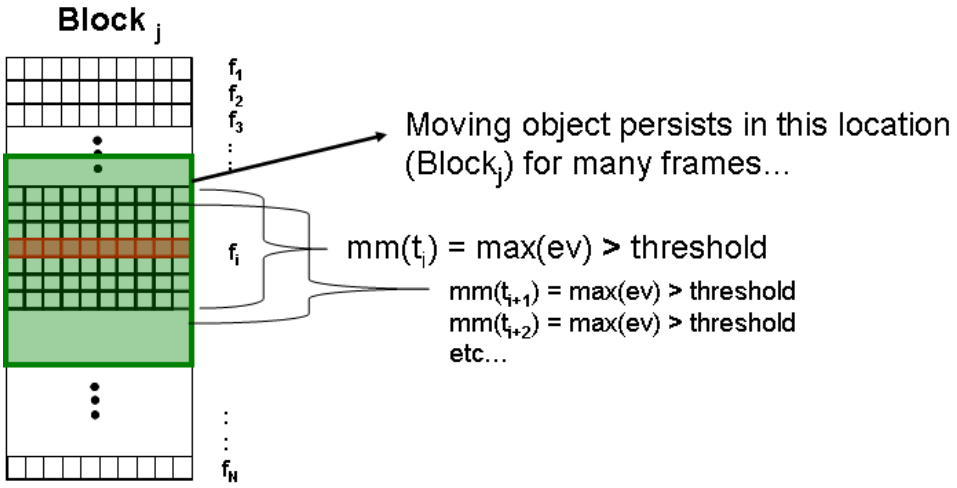


FIGURE 4.5 – Motion Measure for Large/Slow Moving Object.

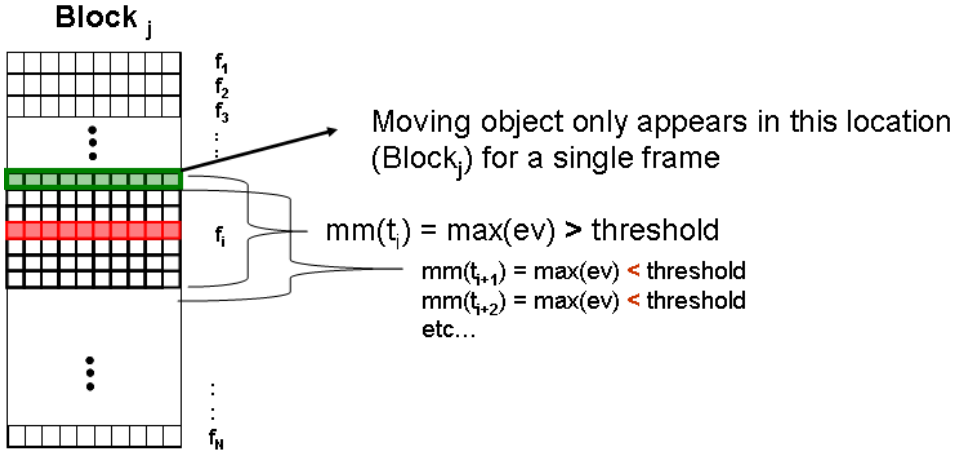


FIGURE 4.6 – Motion Measure for Small/Fast Moving Object.

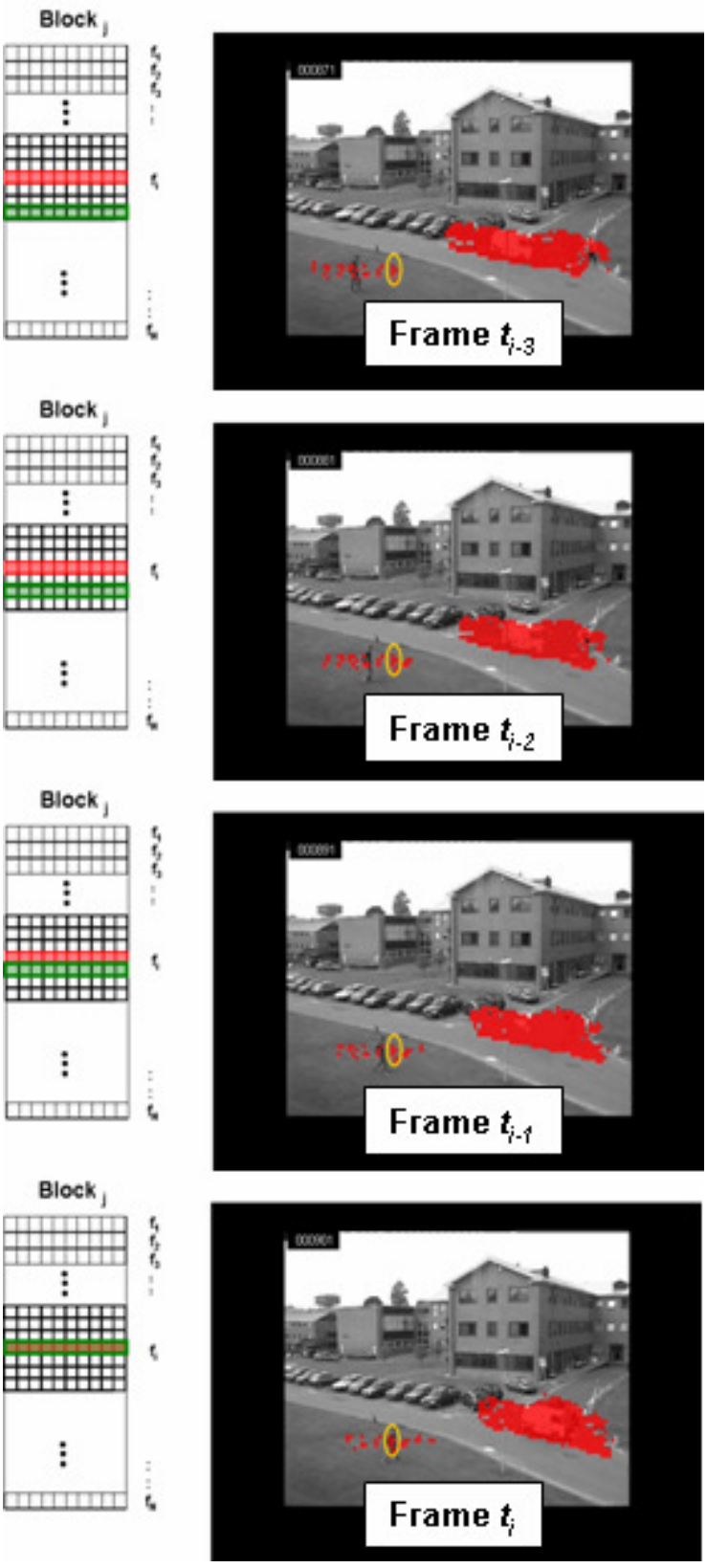


FIGURE 4.7 – Motion Detection Ghosting (Seven-Frame Temporal Window).

However, when a fast and/or small object enters the seven-frame window, it may only appear once (in one temporal frame) and then move on to a different location in the next frame (figure 4.6). In this case, the single frame detection persists throughout the seven-frame window, causing a motion measure large enough to "corrupt" all seven frames at that spatial location. To explain this further, it was helpful to evaluate a single block of pixels over time (figure 4.7). Each frame in the figure shows a given location (circled in yellow). To illustrate the motion measure for the seven-frame window, a graph is shown to the left of each frame. In this example, the motion measure becomes significantly larger three frames prior (t_{i+3}) to the actual moving object aligning with the current time (t_i). In other words, the system is detecting motion three frames before it actually occurs at that spatial location. The progression of the next three frames shows that the actual object finally reaches the middle of the seven-frame window. The same ghosting effect lingers for another three frames (t_{i-3}) with similar results.

Understanding the reason behind the ghost detections led to a means of testing all detected blocks using a spectral filter. As described in detail in the methodology section, a comparison with a background model identified bad detections. In this process, it was deemed necessary to accept a few more missed detections in order to mitigate ghost detections. Essentially, missed detections went up a little while false alarms went down significantly. In fact, due to the additional sensitivity of the detector at multiple bands, more detections (including false alarms) were generated. In this case, the spectral filter had a similar spectral advantage in removing bad detections. Essentially, the multispectral data produced more overall detections, which in turn resulted in more valid detections than the single-band case.

A final note regarding synthetic versus real world data: A single background model was sufficient to represent the entire sequence of DIRSIG frames (4,400 images). However, because WASPLITE data had real temporal variability, the background model became "stale" after a few hundred frames, which resulted in less effective filtering. Thus, the WASPLITE spectral filter was revised to update the background every 50 frames. Once the correct number of PCs was applied to the SP-vectors and ghost detections were removed, valid motion detection results could be evaluated. To get a sense of the improvement using the spectral filter, example results on both DIRSIG and WASPLITE data are presented in the next section

4.2.2.1 DIRSIG Spectral Filter Results

The performance of the spectral filter manifests directly into a reduced number of false alarms (FA), at the expense of additional missed detections (MD). Performance surfaces were plotted for FA (figure 4.8) and MD (figure 4.9) both before and after spectral filtering. As stated earlier, the removal of false alarms due to filtering also included the removal of valid targets that resembled the background model. Despite the desire to reduce the number of missed detections, the trade was acceptable when compared to the advantage of removing invalid (ghost) targets.

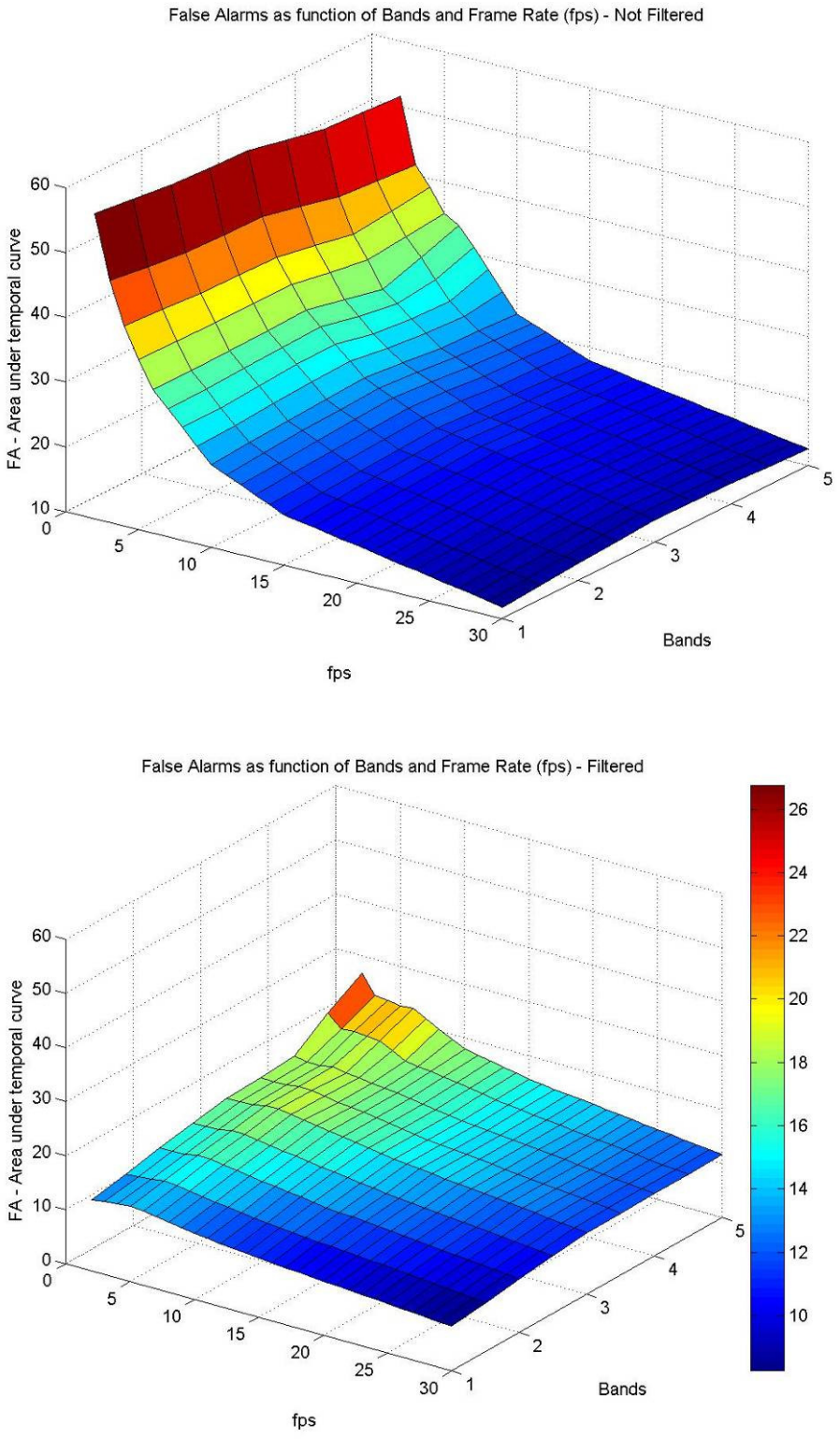


FIGURE 4.8 – False Alarms (DIRSIG Example).

Not Filtered (Top), Filtered (Bottom)

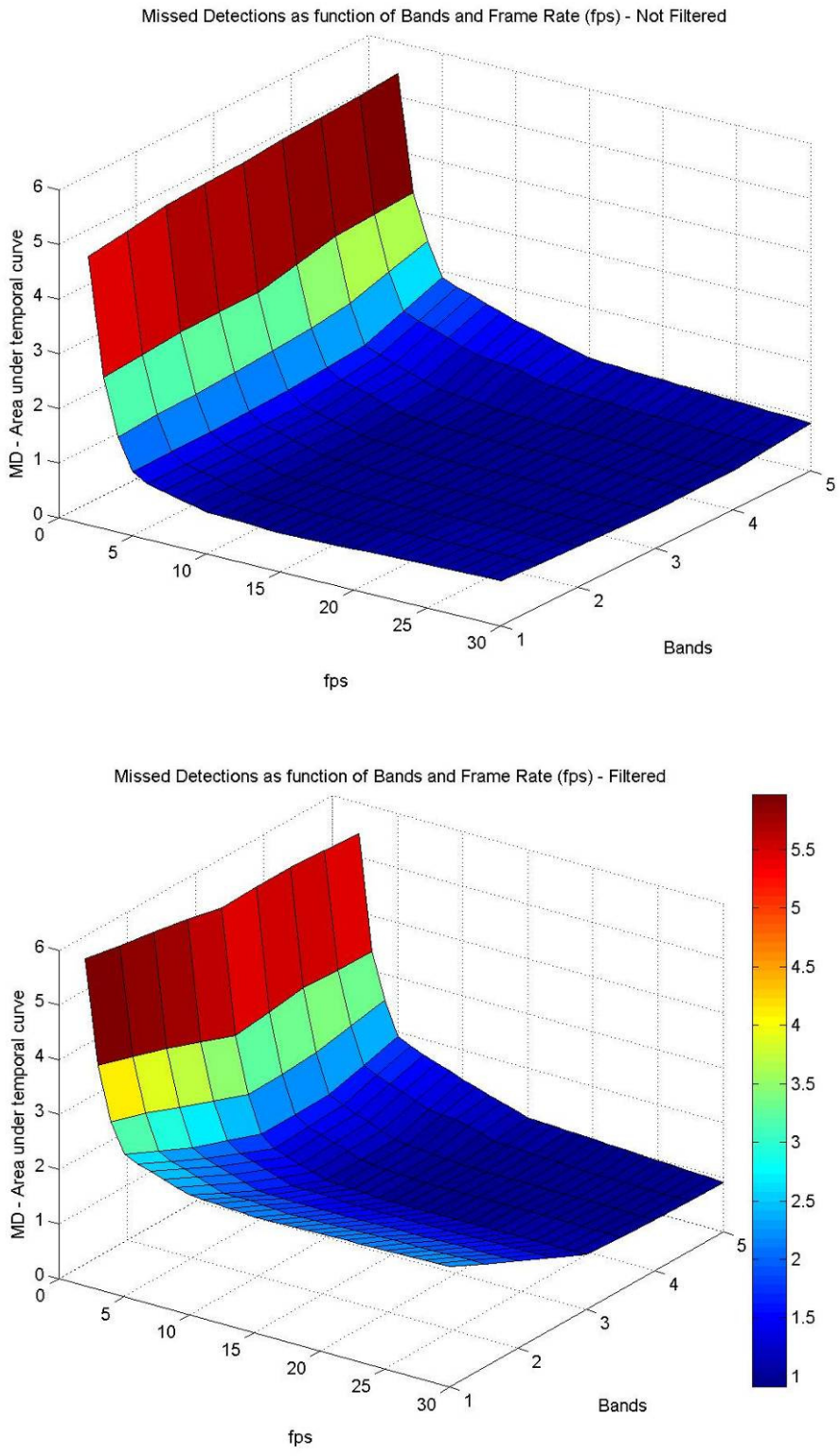


FIGURE 4.9 – Missed Detection (DIRSIG Example).

Not Filtered (Top), Filtered (Bottom)

These surface plots show the expected trend: False alarms are significantly reduced at the expense of slightly more missed detections. The same vertical scale is used in each pair of plots to facilitate comparison. Recall from the metrics section that these values are generated from the area under the temporal curve; hereafter we will refer to them as "area units" (au).

As expected, false alarms due to ghosting were more pronounced at lower frame rates. The maximum value for unfiltered FA is at 1 fps, consistently across the number of bands (figure 4.8, top). However, the filtered maximum FA value (figure 4.8, bottom) has been reduced by over half the amount of the unfiltered plot. The motion truth baseline value for the actual number of moving pixels is about 24 au. Thus, FA has gone from three times this value to nearly one-half that value due to filtering. Notice that the FA values still seem slightly higher in the five-band case; this can only be attributed to poor synthetic modeling of the SWIR band. These reductions in FA are very desirable, especially when considering the moderate increase in missed detections, as seen in figure 4.9.

At the lowest frame rate, the maximum value for unfiltered MD is 5 au, whereas the filtered MD only increased to 6 au. In comparison to the motion truth value of 24 au, missed detections are relatively low and remain low even after filtering.

The synthetic data actually resulted in more false alarms than real world data and filtering had a more dramatic effect because of stability in the background model. The real world data had much more variability both spectrally and temporally, making comparisons with a background model less robust. Thus, improvement in WAPSLITE filtered results was less dramatic than with the DIRSIG results; yet just as essential. Again using the same vertical scale in figure 4.10, we see FA before and after spectral filtering (top and bottom, respectively).

4.2.2.2 WASPLITE Spectral Filter Results

With WASPLITE data, the maximum FA value was nearly 250 au (figure 4.10, top), as compared to the motion truth baseline of a constant 62 au. After spectral filtering (figure 4.10, bottom), the maximum FA was reduced to around 20 for the one-band case and below 100 for the five-band case. The next quandary is why does it appear the single-band case does better?

Recall that the priority of this trade study was to reduce missed detections, even at the expense of more false alarms. As seen in figure 4.11, MD increased after filtering, but not as much in the multispectral data. The maximum MD before filtering was about 14 au (regardless of bands). However, the maximum filtered MD was over 20 au for the single-band case, and just under 14 au for the four-band case. In other words, the spectral filter dramatically reduced FA and slightly increased MD, yet with MD minimized in the multi-band cases.

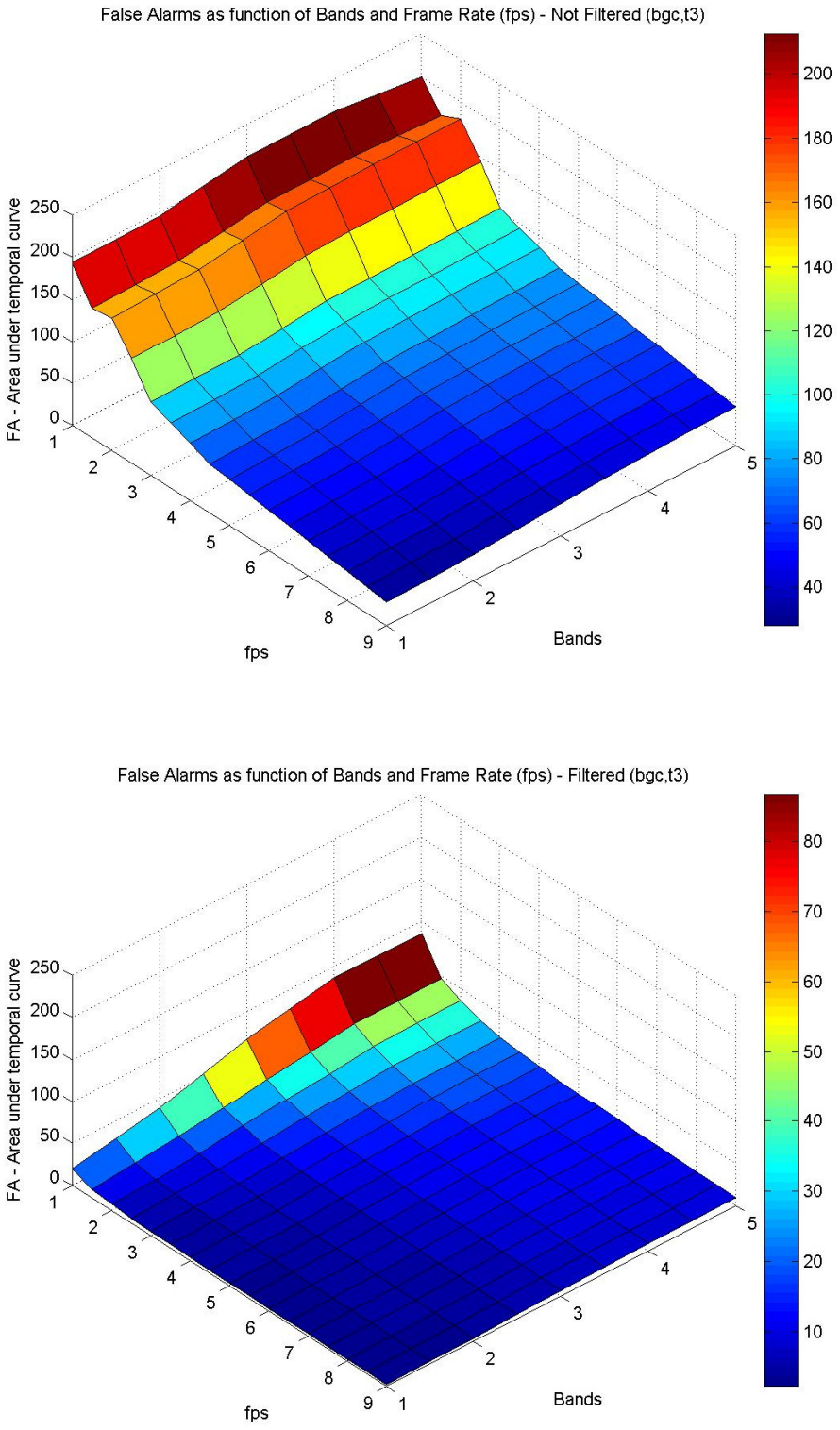


FIGURE 4.10 – False Alarms (WASPLITE Example).

Not Filtered (Top), Filtered (Bottom)

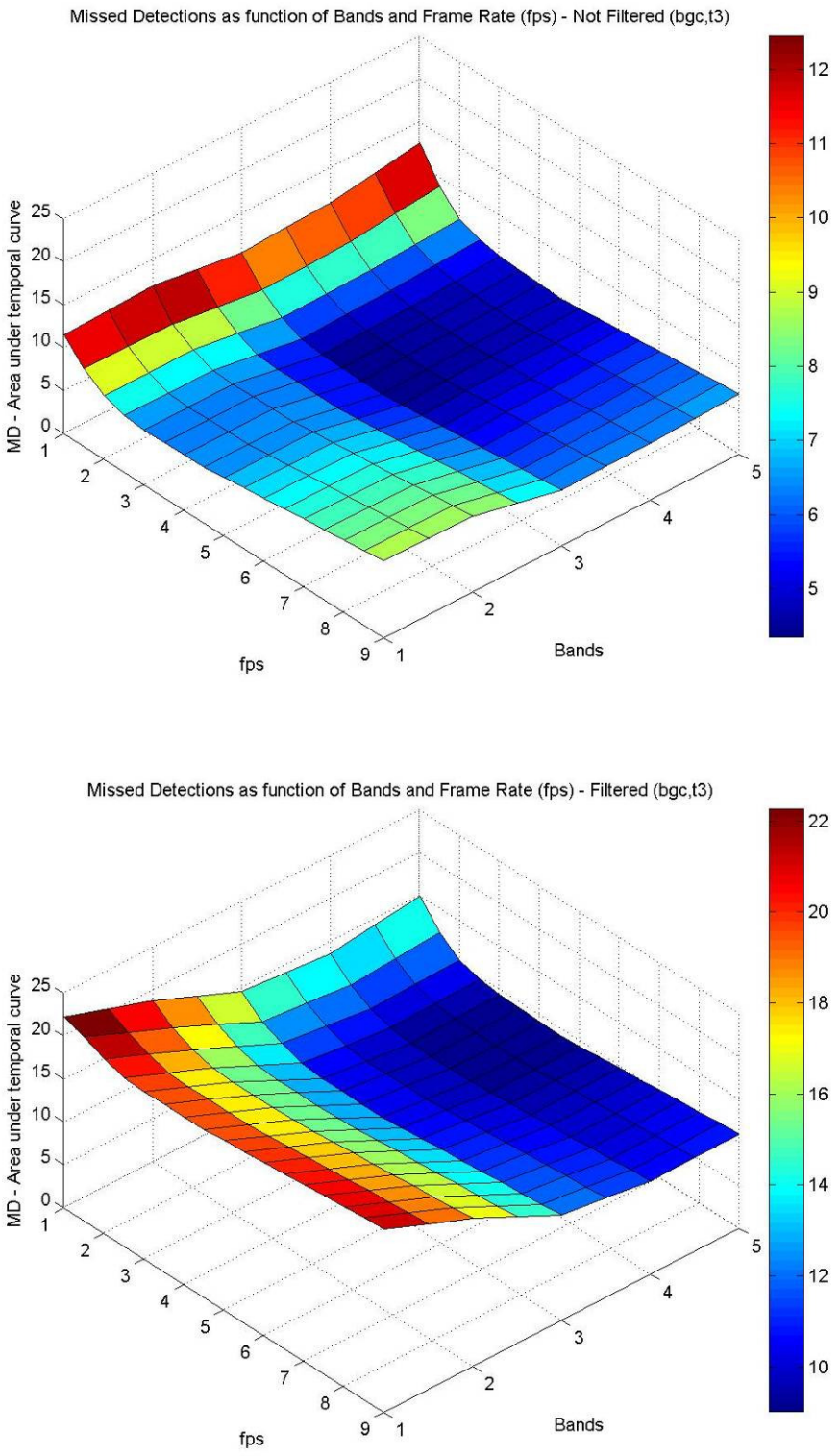


FIGURE 4.11 – Missed Detections (WASPLITE Example).

Not Filtered (Top), Filtered (Bottom)

In summary, filtering the motion detection results by comparing them to a background model assisted significantly in reducing invalid "ghost" detections. However, by removing suspected background pixels, some valid targets were also removed. Although the priority of the surveillance system was to reduce missed detections, the removal of ghosting at low frame rate became the priority. All subsequent data results are assumed to have been filtered and are presented as such. The filtered results provided the detected motion to be further processed by the object segmentation and association functions. The DIRSIG results are presented in the next section, followed later by WASPLITE results.

4.3 DIRSIG Results

The evolution of the motion detection and object segmentation functions began with synthetic DIRSIG data. As a result, much of these performance measures led directly to assumptions (both good and bad) about what to expect in the WASPLITE data. From the start, it was assumed that the synthetic data would be more manageable with less noise—noiseless, in fact, until Gaussian noise was added. However, the results below confirmed some of these assumptions and put question marks after the others.

4.3.1 Motion Detection Results (DIRSIG)

Motion detection results were presented as performance surfaces, as seen in the spectral filtering examples. Figure 4.12 shows the missed detection (MD) results, first as a surface plot (figure 4.12, top), followed by a side-view as a function of frame rate (figure 4.12, bottom). False alarms (FA) are shown in the same fashion in figure 4.13. Although these surface plots are redundant to the spectral filter results, the purpose here is to compare and analyze the filtered results as a function of number of bands and frame rate.

The MD performance surface (figure 4.12, top) clearly shows that using three or more bands results in less missed detections (as desired). Figure 4.12 (bottom) confirms that using three (R, G, B) or four bands (R, G, B, NIR) had the best performance. Surprisingly, these two outperformed the five-band case. Upon inspection of figure 4.13 (top), the FA performance surface provides additional information. As expected, the false alarms were slightly higher in the multi-band cases compared to the single-band case—a fair trade considering the FA values are nearly the same for the one- through four-band cases. However, the 5-band case shows a significant increase in false alarms. The error in the 5-band case can only be attributed to the DIRSIG simulation of the SWIR band; the LWIR simulated band was already removed from the study because it was also found to be unreliable.

Additionally, the performance in all five cases became much worse at very low frame rates (i.e. 1-2 fps); here the MD values increase rapidly. In contrast, MD performance is relatively flat across the bands at frame rates above 5 fps; this was not entirely surprising due to limited variability in the synthetic data. Regardless, the block (pixel) level results showed that multispectral data performed better than the single band case.

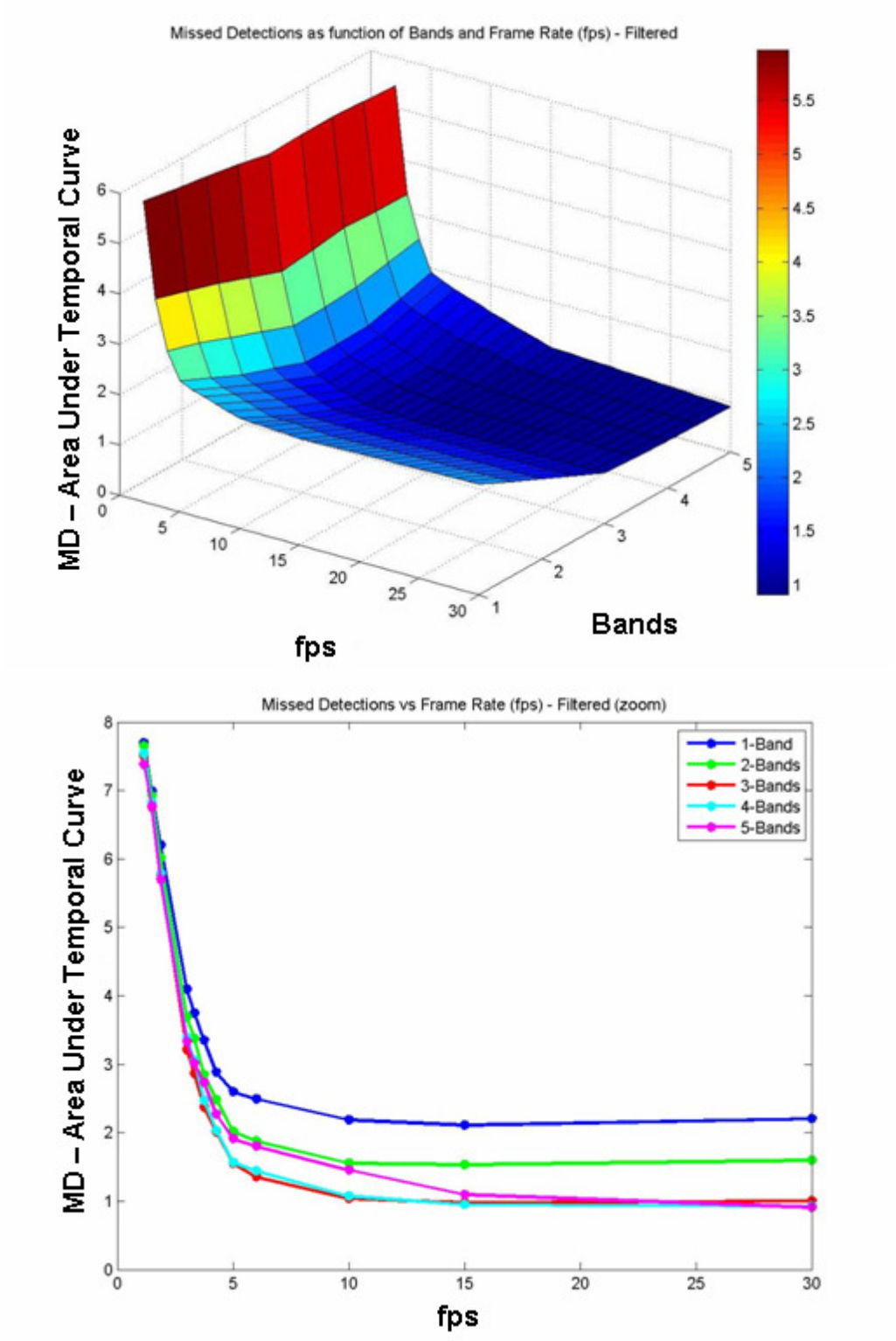


FIGURE 4.12 – DIRSIG Motion Detection Results.

MD as Function of Number of Bands and Frame Rate (Top)
MD as Function of Frame Rate (Bottom)

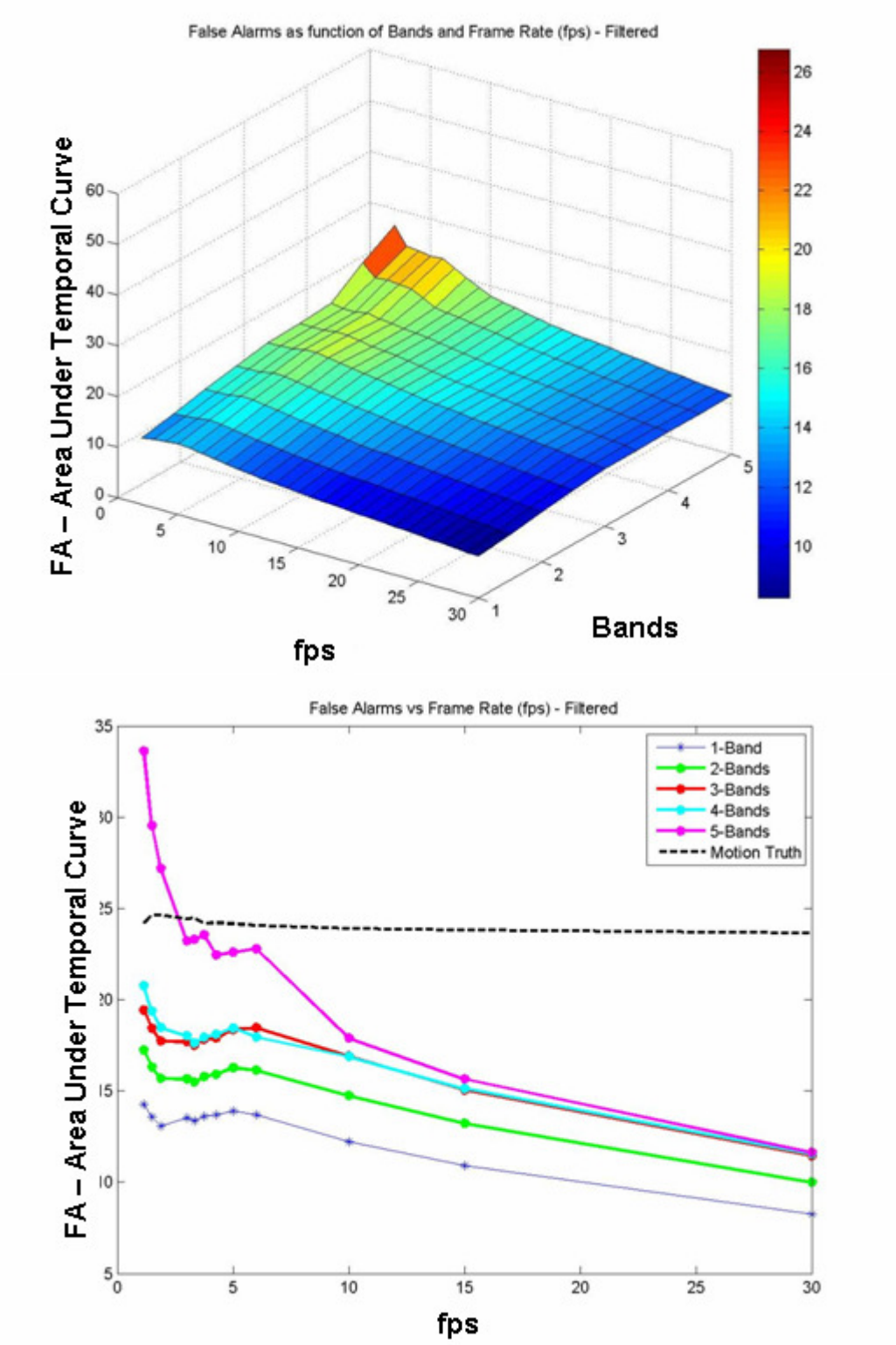


FIGURE 4.13 – DIRSIG Motion Detection Results.

FA as Function of Number of Bands and Frame Rate (Top)
FA as Function of Frame Rate (Bottom)

4.3.2 Object Segmentation Results (DIRSIG)

Object segmentation results follow the same performance surface format as the MD and FA results. Figure 4.14 shows the missed objects (MO) surface plot (top) and side-view as a function of frame rate (bottom). Figure 4.15 shows false object (FO) results in the same fashion.

The expectation was that detected motion blocks (pixels) would be further filtered and collapsed into distinct targets at the object level. Not surprisingly, the performance curves are very similar to the block-level results. Again, we see the single-band case with more missed objects than the multispectral results (figure 4.14); here too we see that the five-band case did not perform as well as the three- or four-band cases. Likewise, the false objects curves were basically the same above 5 fps. Once again, the five-band case indicated faulty SWIR simulation. Interestingly, the four-band case seemed to perform significantly better at low frame rates. However, all five cases show a distinctive drop-off at very low frame-rates, indicating a possible problem with sampling (as performance was not expected to get better at the lowest frame rate).

Overall, the object-level results agree with the block-level results in that "four-bands are better than one" in regards to less missed targets with a small increase in false objects. The single-band case missed about 50% of the moving objects at 30 fps, while the four- and five-band cases only missed about 20% of the valid targets. At lower frame rates, the same trend held, with single-band missing over 60% of the moving objects, whereas the multi-band cases only missed about 50% of the targets. These performance gains came with very little change in false objects at frame rates above 10 fps.

The lack of spectral clutter and sensor noise certainly made DIRSIG detections easier and more robust (as was seen in spectral filtering). However, the same robustness made the performance surfaces somewhat flat in regards to spectral influence. Add to those concerns the uncertainty of modeling the SWIR band, and the results seem a bit underwhelming. However, as will be seen in the next section, the true test of multispectral detection performance depends upon real world results. In addition to the block- and object-level results, object association (or matching) was applied to the WASPLITE data.

In developing the WASPLITE subtasks, algorithm improvements were required to accommodate real world data with additional noise, registration, and other challenges. Consequently, the WASPLITE versions of the software code were significantly streamlined and more robust; especially in the object segmentation subtask. Furthermore, the object association subtask was only applied to the WASPLITE data. Once again, the multispectral detection and segmentation subtasks were expected to outperform the single-band case. More importantly, the ability to distinguish one object from another from frame-to-frame became the final test of overall system performance.

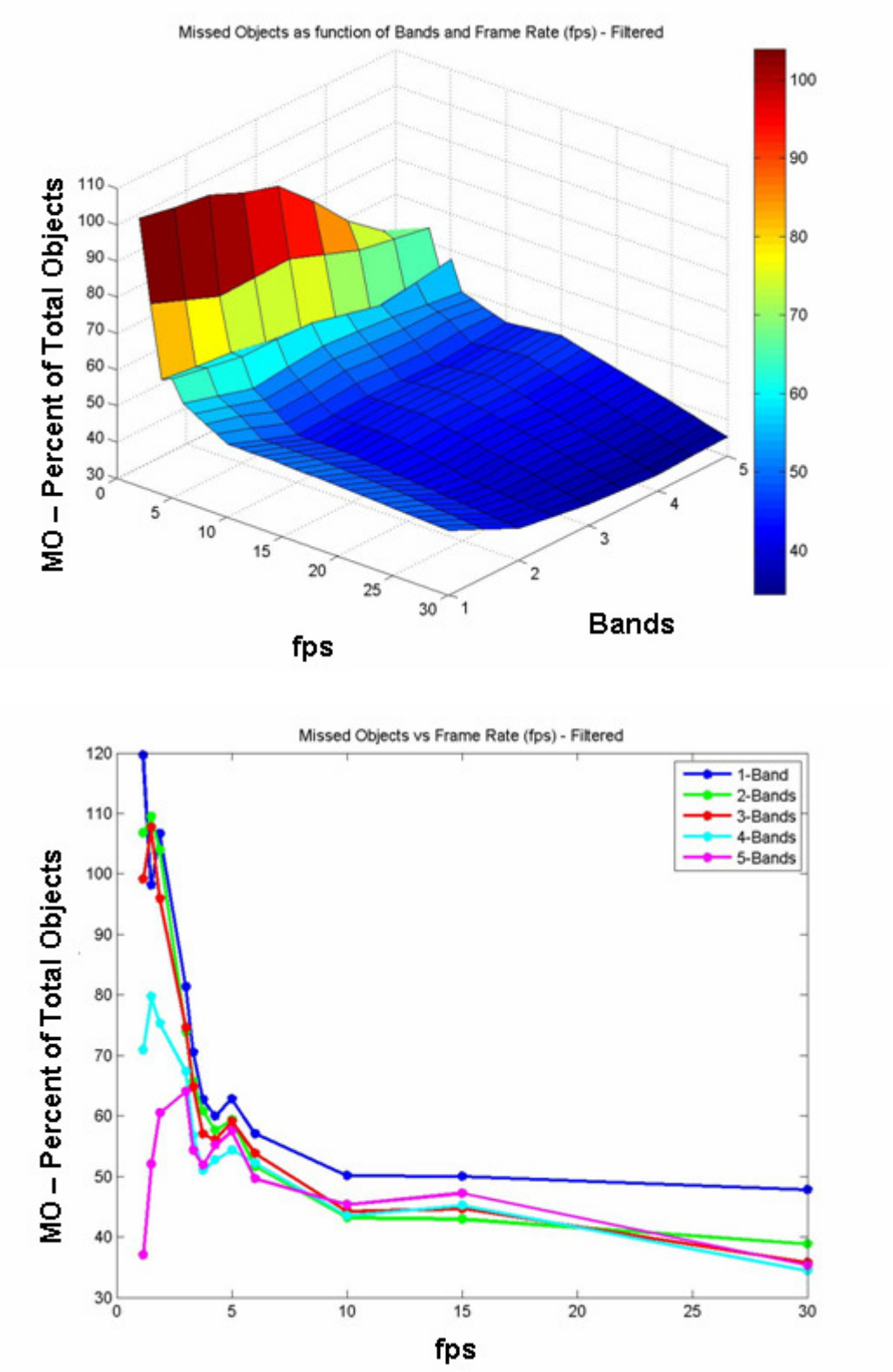


FIGURE 4.14 – DIRSIG Object Segmentation Results.

MO as Function of Number of Bands and Frame Rate (Top)
MO as Function of Frame Rate (Bottom)

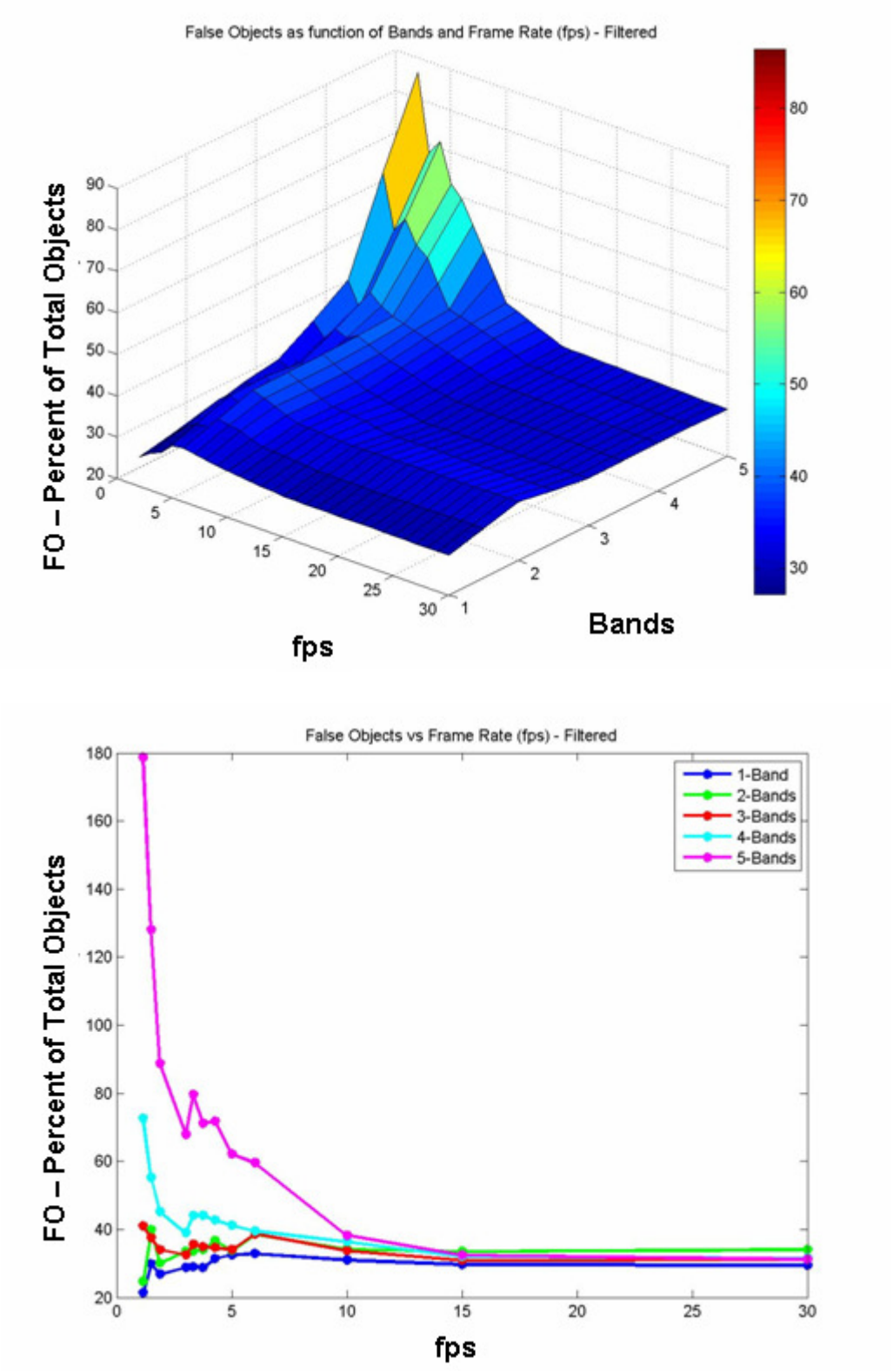


FIGURE 4.15 – DIRSIG Object Segmentation Results.

FO as Function of Number of Bands and Frame Rate (Top)
FO as Function of Frame Rate (Bottom)

4.4 WASPLITE Results

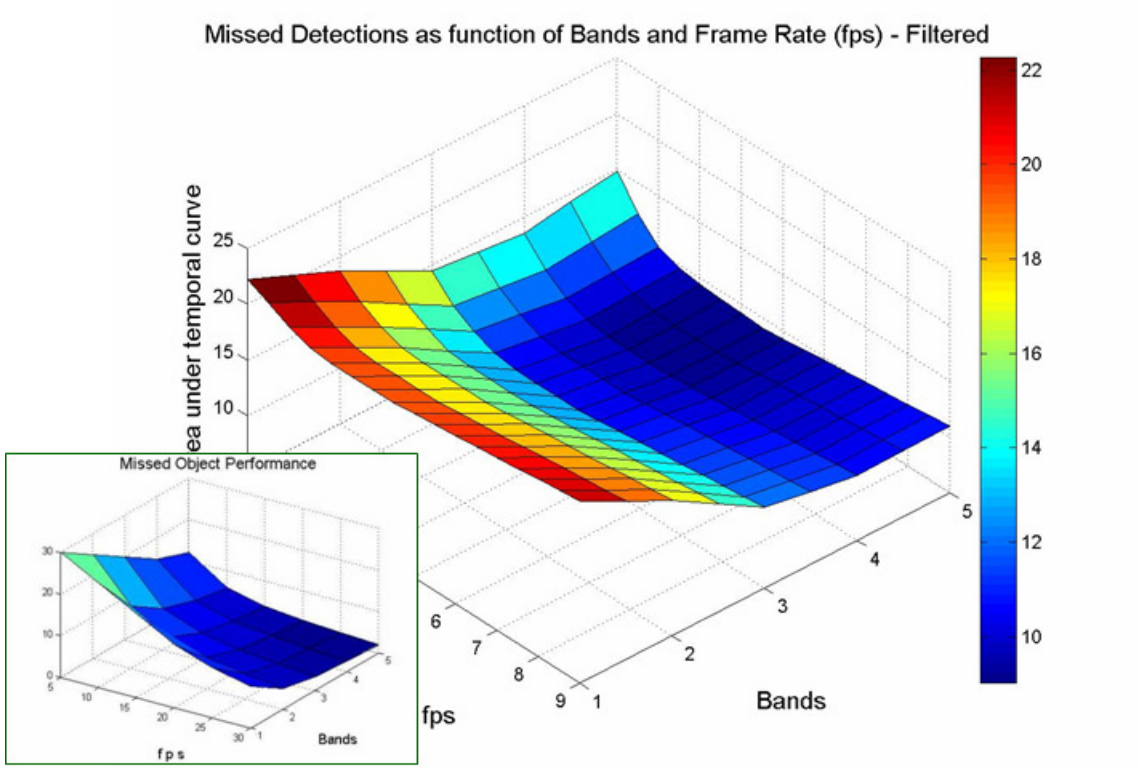
The DIRSIG results were in agreement with the overall hypothesis that multispectral data would perform better than the single-band case using an existing motion detection algorithm. Motion detection and object segmentation both outperformed the single-band case using synthetic data. However, the true test of the multispectral method was achieved using WASPLITE data. In order to accommodate real world problems such as noise, registration error, and a changing background, the entire process was revised and improved. Thus, the final version of the code to perform moving object detection, segmentation, and association was implemented for WASPLITE. The improvements included streamlined pre-processing, an updated background model for spectral filtering, revised thresholding for detection, and additional morphology for object segmentation. Finally, the object association task was implemented to determine the expected spectral advantage in an overall surveillance system.

4.4.1 Motion Detection Results (WASPLITE)

Detecting moving blocks of pixels followed the same basic approach as with the DIRSIG data. However, the settings for spectral filtering and the dynamic threshold were adjusted to suit the new dataset. The results follow the same format as before; missed detections (MD) are shown in figure 4.16 and false alarms (FA) are seen in figure 4.17. The first aspect to note in these results is that the performance surfaces appear smoother and more uniform in contrast to the DIRSIG results. Specifically, the fifth band (LWIR in this dataset) does not appear as erratic as with the synthetic data.

Here again, we see the expected results: Multispectral data outperformed the single-band data. Notably, the five-band performance was equivalent to the four-band case; whereby the four-band case showed a slight advantage in MD at the lowest frame rates (figure 4.16). Another way to interpret these results is to notice that the multispectral cases had less missed detections at the lowest frame rate (1 fps) than single-band did at full frame rate (9 fps). It is also satisfying to see that the notional results (inset in figure 4.16, top) agree quite well with the general shape of the MD performance surface. However, the FA performance surface (figure 4.17, top) shows a proportional increase in false alarms as a function of the number of bands processed. In fact, the increase in false alarms is considerably higher at low frame rates. Whereas the DIRSIG data was rather flat for much of the data, the WASPLITE results show a strong relationship between number of bands and false alarms. At full frame rate (9 fps), all five cases are below 10 au. However, at the lowest frame rate (1 fps), the four- and five-band cases are above 80 au and the single-band case only increase to 20 au. To keep these numbers in context, motion truth was at a constant ~62 au. Thus, we have a specific trade area to consider: Generally flat but improved missed detection performance across the temporal range; however, false alarms go up rapidly at low frame rates.

As stated earlier, the entire process was tuned to allow higher FA in order to achieve the goal of reduced MD. Thus, the hypothesis of better performance at low frame rates (at the expense of additional false alarms) has been verified at the block-level. The next set of results makes the case by performing object-level comparisons.



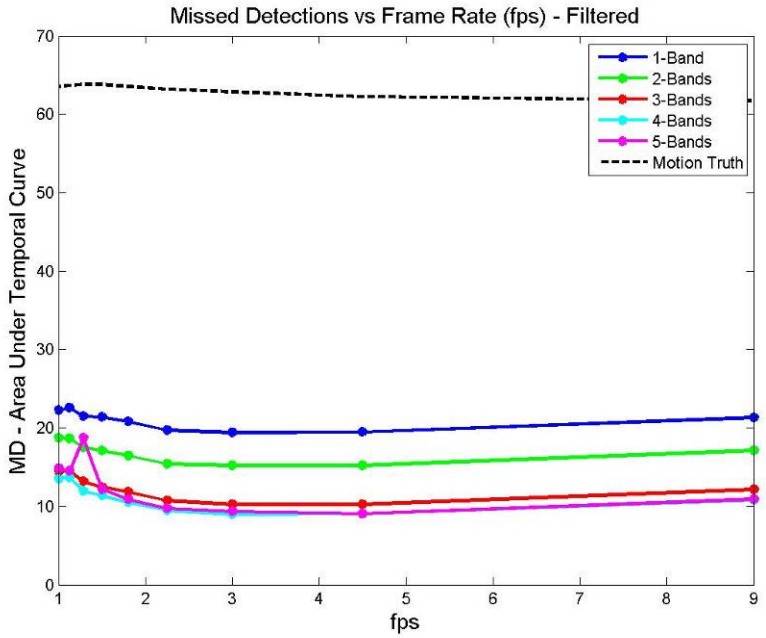


FIGURE 4.16 – WASPLITE Motion Detection Results.

MD as Function of Number of Bands and Frame Rate (Top)
MD as Function of Frame Rate (Bottom)

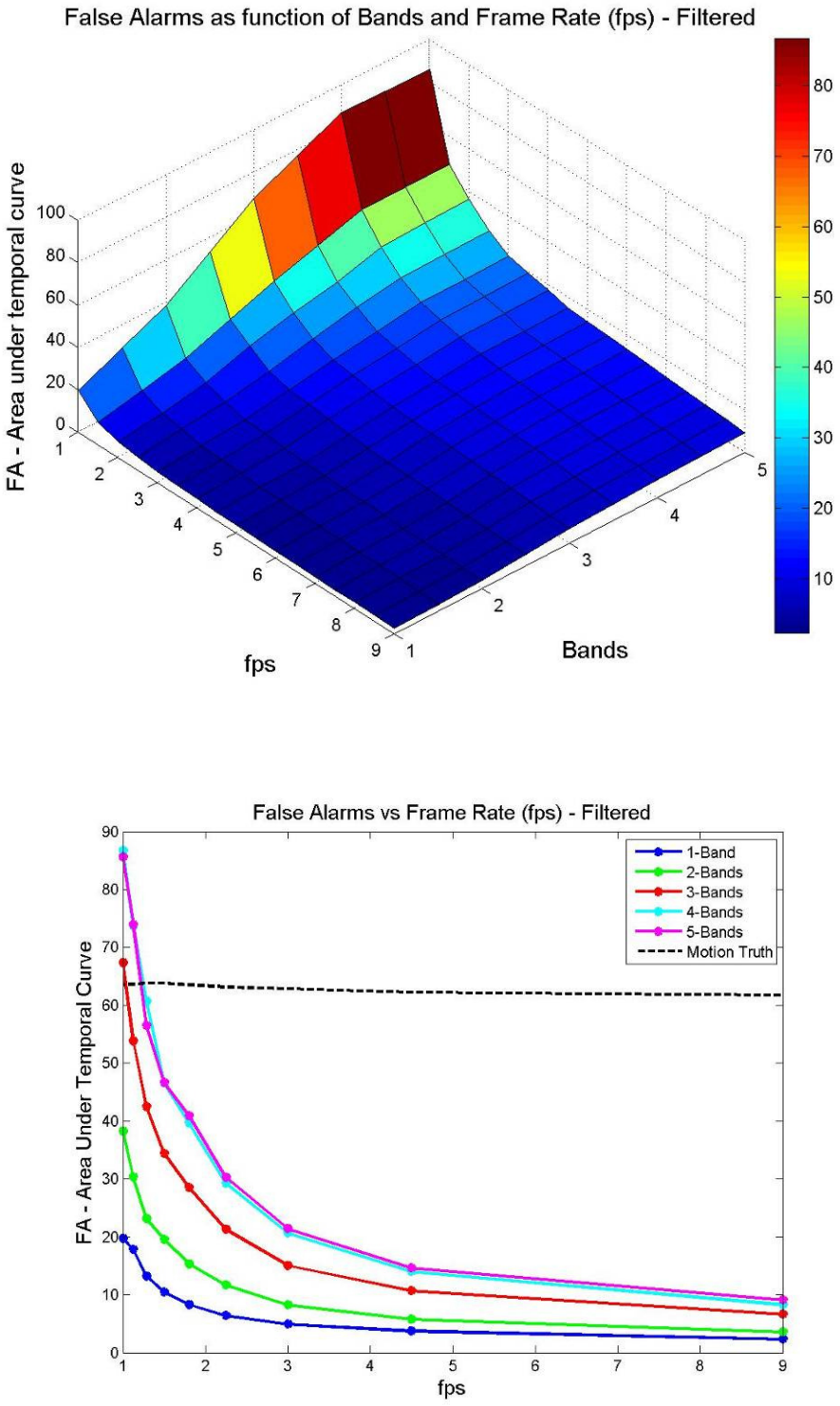


FIGURE 4.17 – WASPLITE Motion Detection Results.

FA as Function of Number of Bands and Frame Rate (Top)
FA as Function of Frame Rate (Bottom)

4.4.2 Object Segmentation Results (WASPLITE)

The object-level results exonerate the block-level results to some extent. Missed object (MO) performance (figure 4.18) was compared to the false object (FO) performance (figure 4.19). Here, the multispectral object segmentation worked much better than for the single-band case. The number of missed objects for the single-band case was around 40% of the total number of valid objects. The best multispectral performance (again, using four-bands) stayed below 20% for most of the temporal range. Again, the multispectral cases at the lowest frame rate (1 fps) outperformed the single band case at full frame rate (9 fps). However, the number of false objects did not go up nearly as much as the block-level false alarms did. Similar to the DIRSIG results, the four-band case outperformed the five-band case. With WASPLITE data, however, the difference in performance can be explained more easily. The fifth band (LWIR thermal bolometer) has nearly one-fourth the resolution of the other four cameras, making the additional band of questionable value. It could easily be adding more noise with less information content, thus degrading performance seen in the four-band case.

Interestingly, the performance in all cases appears to improve slightly for the first two sub-maximum frame rates (4.5 and 2.25 fps). This may actually show the strength of the segmentation process in general. Consider that the highest frame rate case (9 fps) has double the frames of the next highest frame rate case (4.5 fps). Having twice the number of noisy, real world motion frames might actually increase the probability of noisy block (or pixel) detections. However, the segmentation process weeds out these noisy returns, producing more reliable results at the object level.

More importantly, the MO performance was nicely separated (figure 4.18, bottom) as a function of bands, but the FO performance remained similar, regardless of the number of bands (figure 4.19, bottom). In fact, the percentage of false alarms is basically equal down to 4.5 fps, with nearly zero false objects for all cases. However, at lower frame rates, the number of false objects does indeed increase slightly with the number of bands. At the worst case frame rate (1 fps) the increase in false objects becomes more extreme. Nonetheless, MO performance was drastically improved using multispectral data. Furthermore, the notional results (figure 4.18, top – inset) agree quite well with the actual results, once again validating the underlying hypothesis of this study. The object segmentation subtask appears to have made this point more clearly. Now, the final question rests with the object association task: Regardless of how much we have reduced the number of missed objects, how well can we tell the detected objects apart?

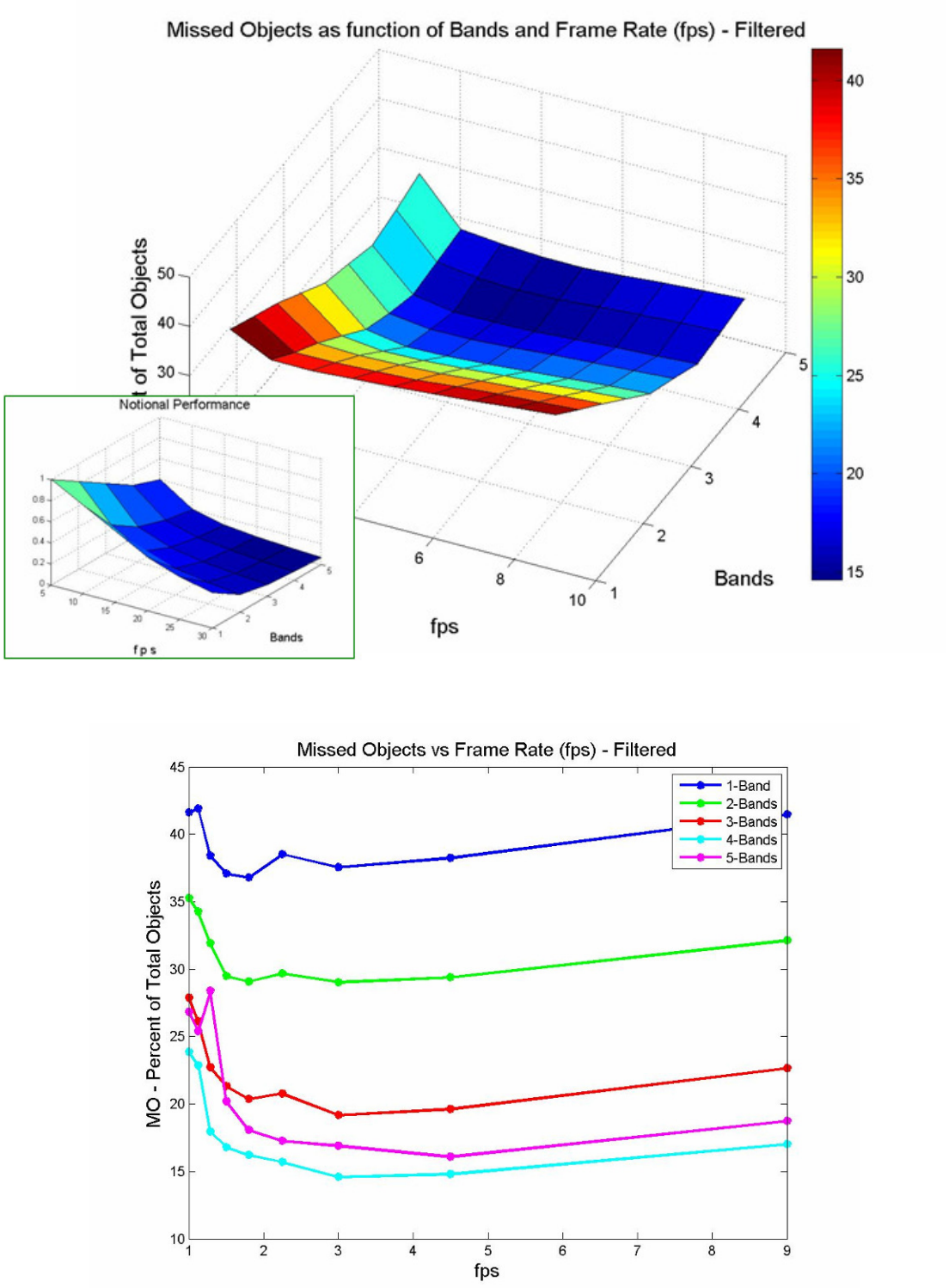


FIGURE 4.18 – WASPLITE Object Segmentation Results.

MO as Function of Number of Bands and Frame Rate (Top)
MO as Function of Frame Rate (Bottom)

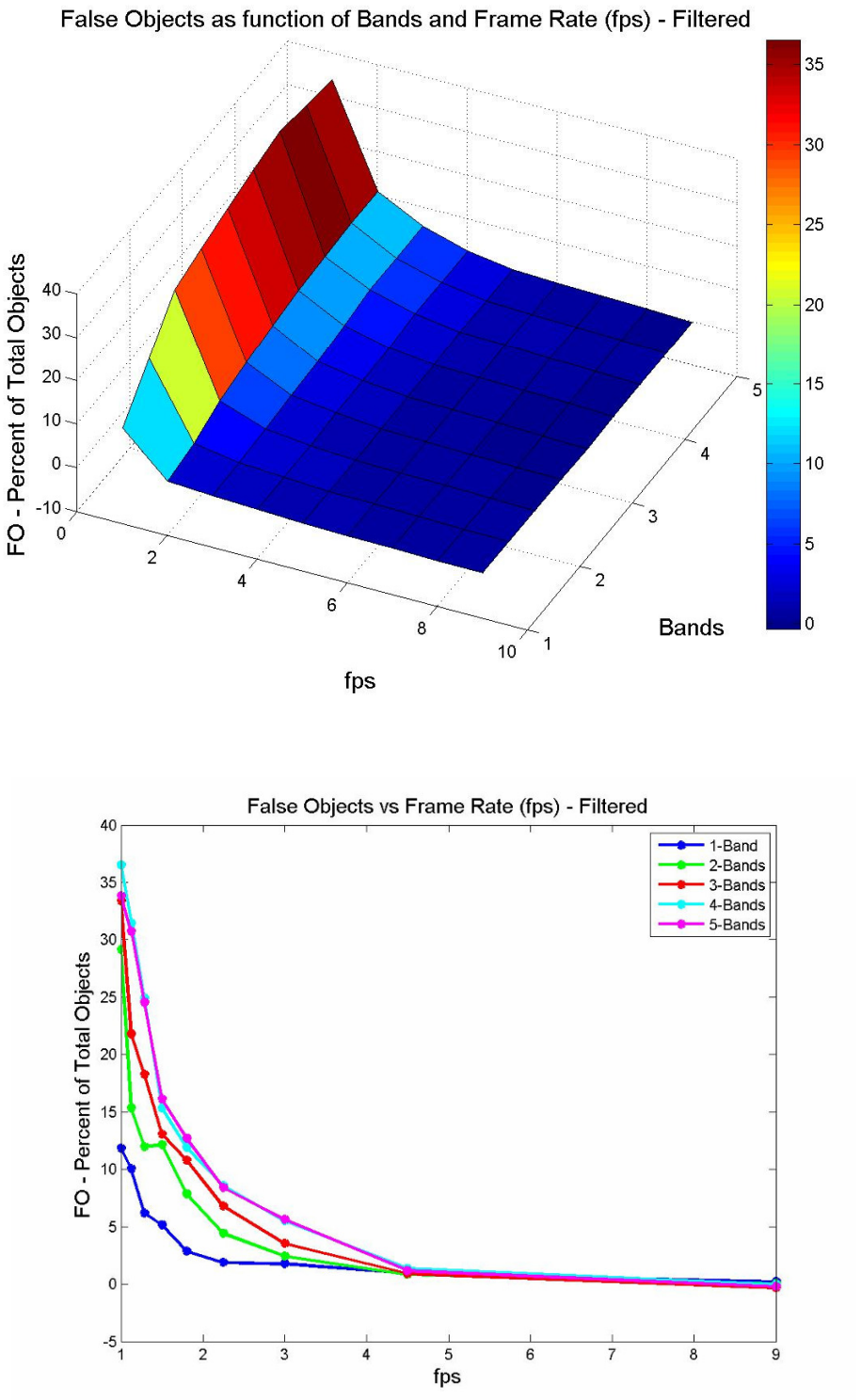


FIGURE 4.19 – WASPLITE Object Segmentation Results.

FO as Function of Number of Bands and Frame Rate (Top)
FO as Function of Frame Rate (Bottom)

4.4.3 Object Association Results (WASPLITE)

The object association subtask—although not fully implemented for every detected object—provided the final piece of evidence to validate the original hypothesis: A multispectral surveillance system can have a distinct advantage over single-band systems, especially at low frame rates. As described in the metrics section, the evaluation of object association (or matching) performance was derived from the ability to distinguish between objects. Given the object separability metric, it was feasible to compare object matching results for all five band combinations at four different frame rates.

Object association performance surface shows a dramatic spectral dependence (figure 4.20, top), with raw separability scores increasing with the number of bands. Recall that the separability score was the Euclidian distance of the difference between the best and the next-best matching objects being compared. More simply, the separability score represents how "easy" it was to distinguish between the correct match and the other candidates. For the sake of presentation of these results, units of separability distance will be referred to as "eu" (Euclidian units). Each pair of objects for every frame evaluated received such a score, making the highest frame rate case (9 fps) a much larger sample set than the lowest frame rate (1 fps) subset; this will become important later in the analysis of separability.

The first evaluation of object association performance used the raw separability scores. The single-band case shows a nearly constant value of 15-20 eu, whereas the multispectral advantage shows twice the separability at values around 40 eu (figure 4.20, bottom). Interestingly, the four- and five-band cases had nearly the same performance; once again making the addition of low spatial resolution (LWIR) data of questionable value. One interesting feature of these results is there appears to be little dependence on frame rate, which actually makes sense. The time between observations changed from 9 frames every second to 1 frame every second. Very little change would be expected in spectral characteristics of the moving objects at a time scale of less than a second. Another interesting feature of these plots is the apparent improvement in separability at the lowest frame rate (1 fps); this could be attributed to an insufficient number of samples (see table 4.1).

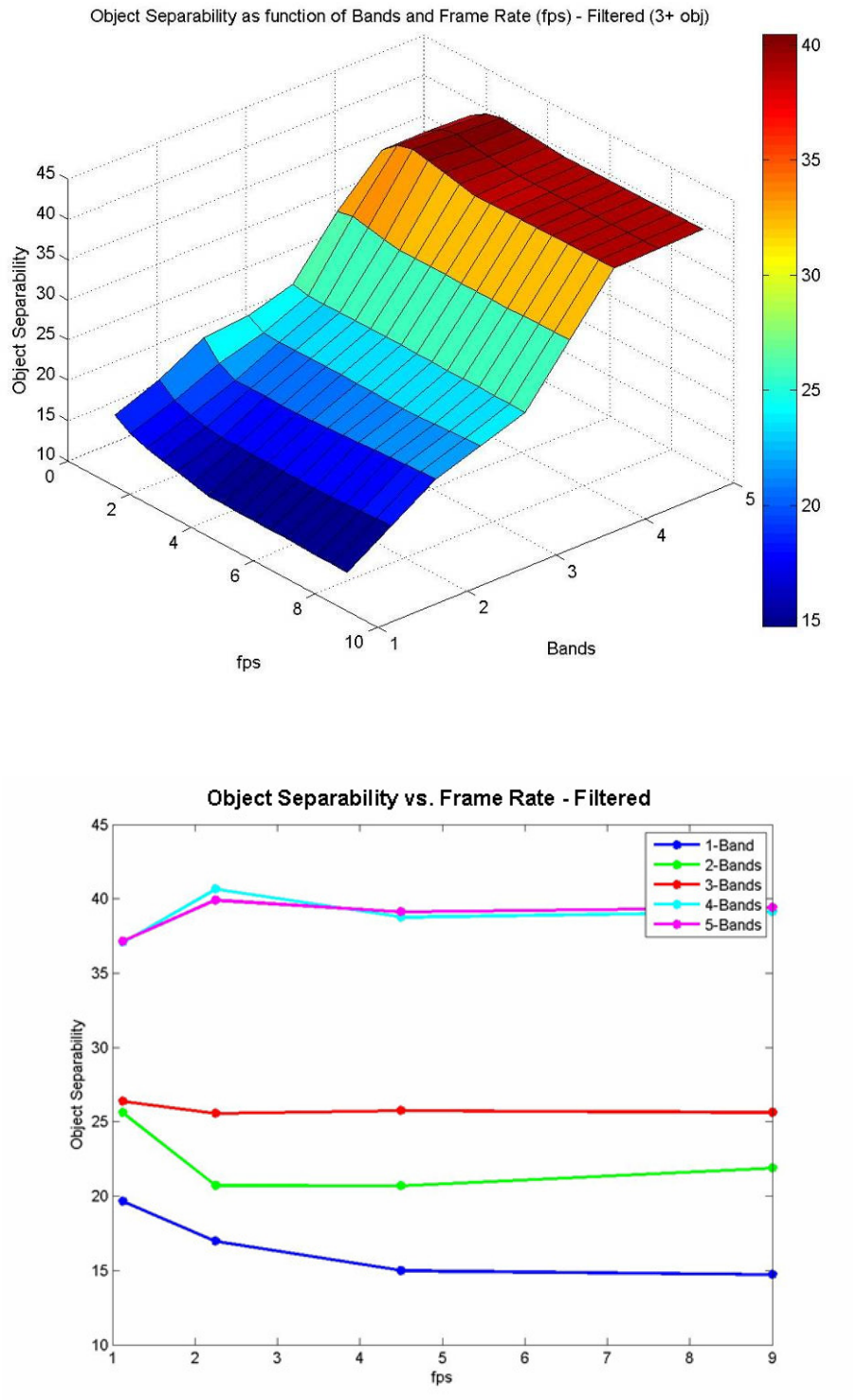


FIGURE 4.20 – WASPLITE Object Association Results.

Separability as Function of Number of Bands and Frame Rate (Top)
Separability as Function of Frame Rate (Bottom)

Evaluating object association performance in the Euclidian difference space (eu-distance) may not have been a fair comparison. Recall that separability scores were the result of the difference between the best and the next-best match. This could be considered as a difference vector between the two measurements (figure 4.21). The magnitude of the difference vector gives us a separability measure for that particular target object.

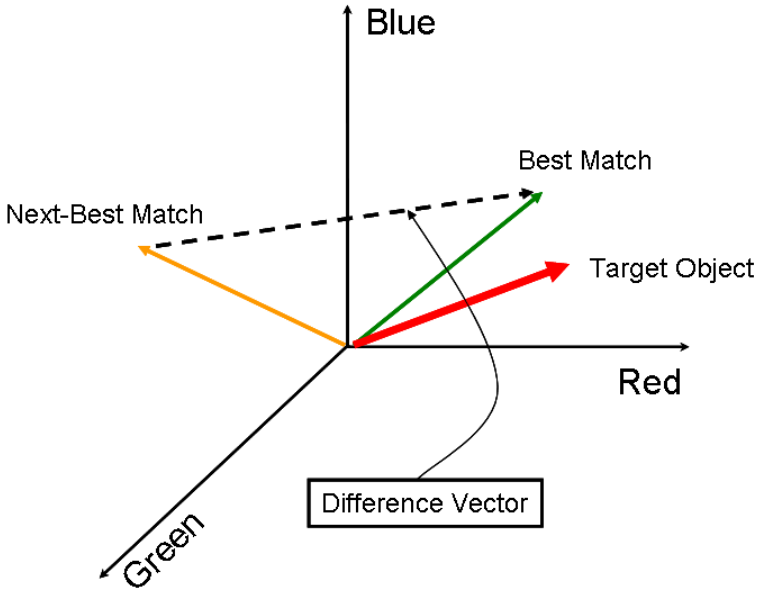


FIGURE 4.21 – Difference Vector Between Best and Next-Best Match.

Because the distribution of each separability histogram was different for each case (band/frame rate combination), the dimensionality of the difference vectors was not necessarily the same. Thus, the magnitude of the difference vectors might not be to scale. A normalization of some kind seemed to be in order. One way in which this was considered was to normalize by the standard deviation of each histogram of separability values. In this way, each histogram is now expressed in numbers of standard-deviations rather than the raw eu-distances. The results of this type of normalization are seen in a revised performance surface (figure 4.22).

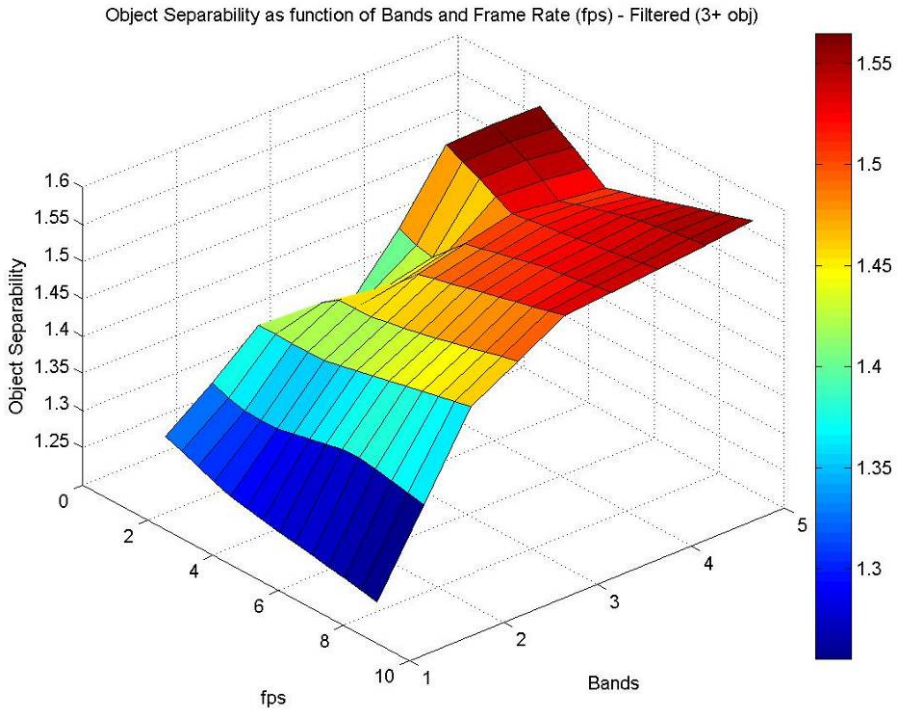


FIGURE 4.22 – Object Separability (Normalized).
(Frame Rate = 9, 4.5, and 2.25)

Neither raw nor normalized separability scores describe the full potential of the spectral advantage. In order to further demonstrate the value of multispectral data, the normalized separability histograms (figure 3.43) can be viewed in another way. Consider an operational system that cannot distinguish between two objects if the normalized separability is too small. For example, assume a surveillance system needs a separability of greater than 0.5 σ to achieve object association. If two objects are not distinguishable, both will need to be tracked until the ambiguity is resolved. Figure 4.23 shows two normalized separability histograms with a threshold at σ = 0.5. The single-band histogram (left) shows that 27.7% of the objects would fit this category, while the four-band case shows only 15% of the objects would be indistinguishable.

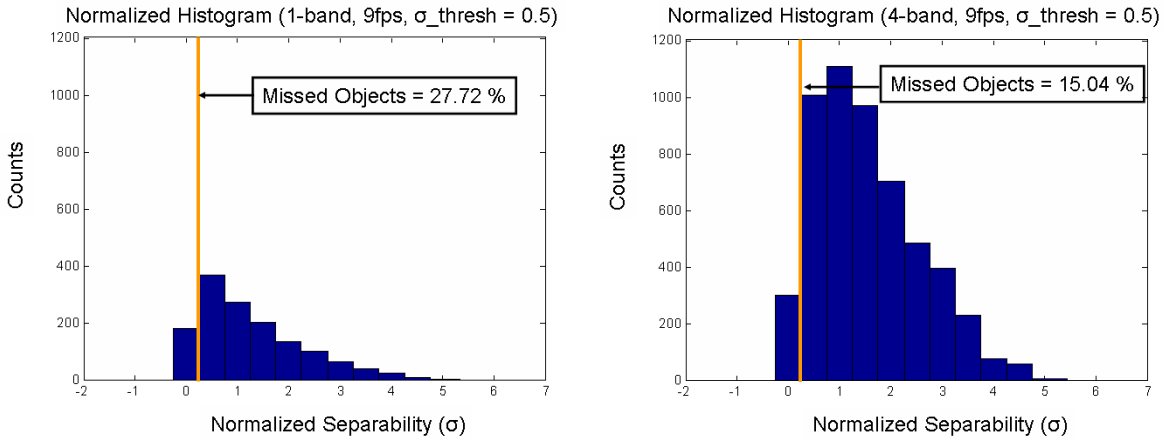


FIGURE 4.23 – Normalized Object Separability (Threshold = 0.5 σ)

Similar to the other performance surfaces, the percentage of missed objects can be seen as a function of the number of bands and frame rate in figure 4.24. Here we see the number of missed (or indistinguishable) objects as a percentage of the total number of objects compared. At maximum frame rate, there is an advantage at three-bands. However, at lower frame rates, the four- and five-band cases show the best separability (i.e. the least number of missed objects). Although the performance surface is not directly associated with frame rate or number of bands (i.e. not very smooth), the general trend shows significantly less missed objects as a function of more spectral bands.

The advantage changes as a function of frame rate partially due to the fact that more bands generally produced more detected objects. More detected objects provided more objects to compare, as can be seen in figure 4.23. The single-band histogram has less overall objects (counts) as compared to the four-band histogram. A summary of the number of objects for each case can be seen in table 4.1, where more bands resulted in more objects available for comparison.

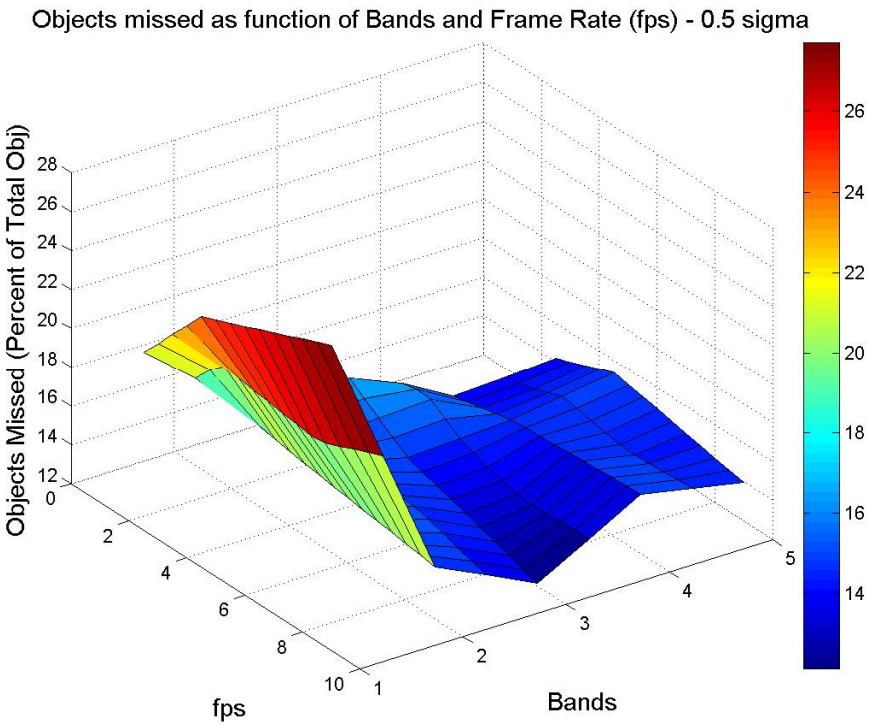


FIGURE 4.24 – Missed Objects Due to Insufficient Separability (Threshold = 0.5 σ)

Another important aspect of evaluating separability was in considering that observations at lower frame rates were a subset of the higher frame rate samples. In fact, there were significantly fewer objects evaluated at the lowest frame rate (1.13 fps) than at higher frame rates, as summarized in table 4.1. The multispectral data at the highest frame rate has significantly more samples than the single-band case.

	9 fps	**4.5 fps**	**2.25 fps**	**1.13 fps**
1-Band (G)	**1396**	**518**	**169**	**27**
2-Bands (R,G)	**2990**	**1137**	**393**	**77**
3-Bands (R,G,B)	**4022**	**1645**	**502**	**91**
4-Bands (R,G,B,N)	**5331**	**1996**	**586**	**100**
5-Bands (R,G,B,N,L)	**5273**	**2021**	**570**	**113**

TABLE 4.1 – Number of Associated Objects at Each Band/Frame Rate Combination.

Because the number of samples was so low in the last case (1.13 fps), the normalized performance surface was shown with only the highest-three frame rates (figure 4.22). Performance degrades rapidly using less than four bands due to insufficient sampling. The four- and five-band cases show the expected trend of improved separability as a function of the number of spectral bands processed.

In summary, the performance results for both datasets have validated the original hypothesis. Generally, using more spectral bands provided better performance. Initially, more bands resulted in more detected motion, which in turn resulted in better object segmentation. Given a larger number of objects to compare, the object association task on real world data indicated best performance using all five spectral bands of information. Additionally, the results validated the secondary objective of evaluating reduced performance due to low frame rate data. Poor single-band performance was mitigated by using additional bands.

Chapter 5

Summary

5.1 Conclusions

The objectives of this project were successfully accomplished. Both synthetic and real world data were evaluated to test the hypothesis that using more spectral bands would provide better moving object detection, segmentation, and association performance. A methodology was established to evaluate performance as a function of spectral and temporal resolution, which enables a trade study to be performed on any such datasets. The performance results can be considered as two separate conclusions which support the hypothesis of a spectral advantage in a persistent surveillance system.

The first piece of convincing evidence appeared in the motion detection and segmentation results. When applied to the operational example presented in the introduction (figures 1.1 and 1.2), a multispectral sensor with a spinning mirror could conceivably cover 9 times the area of responsibility (AOR) at only 4 times the bandwidth. In the case of WASPLITE, multispectral moving object detection and segmentation performance at 1 fps was equivalent to (or better than) single-band performance at a full frame rate of 9 fps. Thus, a multispectral sensor could spend one-ninth the time on each area of interest (AOI) and get the same moving object detection performance as single-band surveillance focused on only one AOI.

The second example of the spectral advantage can be seen in the object association results. Although not implemented in an end-to-end system, the object association experiment provided confirmation that there is a significant spectral advantage. Object matching was considered the final subtask prior to tracking moving objects. Tracking logic would then be applied to the results of matching detected objects from one frame to the next. In the case of object ambiguity, spectral separability has a distinct advantage over a single-band system.

Overall surveillance system performance was assumed to be directly related to the three subtasks evaluated: Moving object detection, segmentation, and association. As such, the methodology provided performance results at each stage of the process: block (pixel) level detection performance in the form of missed detections (MD) and false alarms (FA); object level segmentation performance in the form of missed objects (MO) and false objects (FO); and global performance based on spectral separability in the final object association subtask. When considered as a series of data filters tuned to reduce missed detections, the multispectral cases captured more moving objects than the single-band case. Although the trade study was limited to spectral and temporal resolution, the methodology developed would easily enable a further study on the effects of spatial resolution, additional noise, registration error, and a full-up object matching scheme to combine spatial and spectral information. These topics are discussed in more detail in the future work chapter.

5.2 Contributions

Although the results of this study were satisfying, the true value of the work accomplished was in the amount of learning that transpired. The first task was to find and adapt a state-of-the-art motion detection algorithm. The algorithm chosen was successfully adapted to multispectral data and a novel methodology included a means of testing data at variable frame rates. The inherent low frame rate problem in the original algorithm was solved using a spectral filter. Original object segmentation processing used morphology to remove noise and define the spectral mean of each object. A novel object association metric was developed to demonstrate the spectral advantage when comparing objects from one frame to the next.

Great value was found in developing the datasets to support this study. Developing both synthetic multispectral DIRSIG movies with perfect motion truth and collecting real multispectral WASPLITE data was an essential and nontrivial task. A new method for defining motion truth for the WASPLITE data was also provided via spectral filtering and morphological processing. Further, the recent WASPLITE data registration technique [McNamara:2007] was validated in this project. Both of these datasets are unique and should be utilized in future dynamic imaging projects.

The methodology, metrics, and results of this study should be considered as original work in a field rife with research activity. Multispectral persistent surveillance will be an area of intense effort in the near future, especially in regards to full-time (24/7) operations.

Chapter 6

Future Work

As a first step in dynamic imaging for the Center of Imaging Science (CIS), this multispectral surveillance study provides a platform for future research. To further investigate the trade space, additional items should be considered, including noise and registration error, threshold settings, spatial resolution, future datasets, and object tracking.

6.1 Noise

Signal-to-noise (SNR) on the DIRSIG data was degraded by adding noise artificially. Similarly, registration error was considered to be a potential "show stopper" for this experiment. Fortunately, these sources of error did not impede the successful completion of this project. These variables can easily be degraded artificially in the DIRSIG data and in the best case WASPLITE datasets. In future, noise sensitivity should be addressed for both datasets.

6.2 Threshold Settings

One area of this study required some amount of "trial and error" in setting various thresholds to achieve the desired detection and segmentation results. Further effort to optimize these settings would, once again, be based on user requirements (i.e. false alarm tolerance vice missed detections). Specifically, the dynamic threshold settings (C_1, C_2, u) might be adjusted dynamically according to specific dataset characteristics and performance. However, optimization was considered outside the scope of this project.

6.3 Spatial Resolution

Spatial resolution is a fundamental trade space parameter regardless of spectral or temporal resolution. As such, it becomes a design feature worth investigating in association with the other two variables. Specifically, the current methodology allows for a variable detection window. The detection window for this project was set at constant (4 x 4) pixels. However, this could be reduced to using (3 x 3), (2 x 2), or even (1 x 1) windows. An important note here is that a (1 x 1) window would be impossible using single-band data because the number of pixel observations over three temporal frames would produce an insufficient spatiotemporal vector. Multispectral data might have another advantage in reducing the window size and maintaining sufficient variability to apply this detection technique.

As a practical matter, adding thermal bands to a system inherently incurs a reduction in spatial resolution due to limitations in current thermal detector technology. Some basic spatial resolution topics regarding WASP and WASPLITE data collections are addressed in Appendix A.

6.4 Performance Metrics

For the purposes of this study, performance metrics provided a top-level assessment of the trade space. The goal was to provide a general comparison of system performance as a function of spectral and temporal resolution. However, other moving object detection metrics are available (as described in the background section). Object centroid error would be the first metric to add to future studies. Matlab code for this project has already incorporated place-holders for any number of object features in addition to the spectral mean, including centroid location, object size, identification labels (i.e. car or pedestrian), and so forth. The utility of these object features would be dictated by tracking logic and user requirements.

Data metrics and statistics could be further developed and refined given sufficient statistical information (i.e. we need more data!) A second generation DIRSIG video was proposed over the course of this project, but never came to fruition. (In fact the DIRSIG video generated for this study took several months alone to develop). The next DIRSIG movie should be based on Megascene (Tile 1), as describe in more detail in Appendix B.

6.5 Object Association

Finally, the datasets used for this study were limited to about 4,000 sequential image frames. Thus, the lowest frame rate achievable with a sufficient number of images was about 1 fps. Another uncharted area of motion detection lies within very low frames per second (VLFPS) regime. Datasets with a sufficient duration of activity could investigate extreme cases such as one frame per minute or longer. Here, it is reasonable to assume that the spectral advantage in object association would vastly improve overall tracking performance.

A multispectral tracking system might be able to intermittently lose and then reacquire erratic, unpredictable VLFPS targets. Although not implemented for this project, a novel approach to object association—using both spectral and spatial information—is presented in Appendix C.

Appendix A

Variable Spatial Resolution

As mentioned in the thesis, spatial resolution is an inherent trade when adding infrared bands because infrared detectors tend to require physically larger pixels. As an example, figure A.1 shows typical spatial resolution associated with visible and IR channels in the WASP sensor [WASP:2006]. In this case, the short-, mid-, and long-wave IR channels have a GSD of about 48 inches in comparison to the visible GSD of 8 inches (best case). Thus, the IR channels suffer a degradation of about one-sixth the spatial resolution of the visible bands.

The same is true for the WASPLITE data, where the visible camera images are roughly (640 x 480) pixels, whereas the thermal bolometer produces (320 x 240) pixel images. The LWIR images are one-fourth the spatial resolution of the camera images. An example of the WASPLITE images is seen in figure A.2. Once registered [McNamara:2007], all of the images were reduced and cropped to be (256 x 200) pixels. As mentioned in the methodology section, the reduction of all the images manages to average out some of the difference in spatial resolution. Furthermore, when the motion detector monitors a (4 x 4) pixel window, the final spatial resolution for processing is actually (56 x 50) blocks.

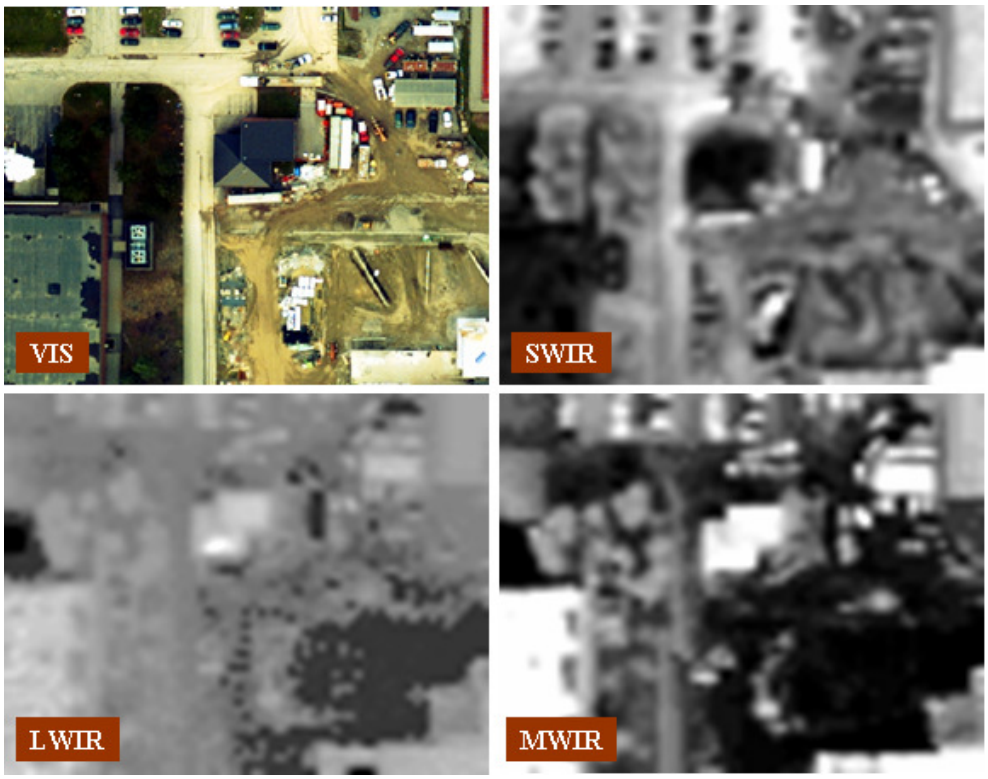


FIGURE A.1 – WASP Spatial Resolution Comparison.

FIGURE A.2 – WASPLITE Spatial Resolution Comparison.

Using the DIRSIG synthetic dataset, system performance was evaluated as a function of spectral and temporal resolution. It would be a simple matter to spatially average the datasets even further to degrade spatial resolution. The DIRSIG data was generated with a best case GSD of 6 inches in all six bands, thus spatial resolution can be degraded incrementally using spatial averaging. In the case of WASPLITE data, the same degradation could be applied.

Finally, one feature of the motion detection routine that should be exploited is the variable size of the block of pixels being monitored. The author of the technique [Miezanko:2006], used a default window of (8 x 8) pixels. This trade study used (4 x 4) pixel windows. Future experiments could be conducted using (2 x 2) or even single-pixel windows, thus preserving the inherent spatial resolution of the source data. Multispectral data is expected to have a significant advantage simply because there are multiple brightness values per pixel. Consider the single-band case using a (2 x 2) window; the spatiotemporal vector (SP-vector) will only have 12 values over three temporal frames (vice 48 values using a (4 x 4) window). Monitoring a single pixel (i.e. an SP-vector with 3 values) would not work. However, given enough bands a single multispectral pixel could have sufficient variability to enable the motion detection technique to work. Essentially, multispectral data should enable processing higher spatial resolution data than a single-band data. The final caveat on this experiment would be to accept the additional processing load for monitoring more blocks per frame.

Appendix B

Second Generation DIRSIG Movie

An improvement on the fidelity of spectral complexity was proposed by a second generation DIRSIG movie, this time based on a subsection of the Megascene Tile 1 [DIRSIG:2006] as seen in figure B.1. A higher fidelity synthetic model would allow for more realistic spectral and spatial clutter. Future DIRSIG movies would provide perfect motion truth and the flexibility to test different sensor combinations in the trade space.

FIGURE B.1 – Second Generation DIRSIG Video.

In this case, the background spectral content would be based on actual ground measurements and overhead photography. Thus, spectral and spatial texturing would be modeled to more closely resemble the real world environment. However, the second generation DIRSIG video was not found to be within the scope or schedule of this project.

Appendix C

Object Association

Conventional single-band methods compute only a spatial correspondence between two grayscale objects, often weighted using a spatial confidence scale as discussed in the background section. Using multispectral data, it seems possible to first compute a spectral similarity score between two objects before using spatial comparisons. The spectral similarity score could even be weighted by a spectral confidence score.

C.1 Combined Similarity Score

After the multispectral data is reduced to a grayscale space, conventional single-band spatial matching provides a second similarity score—also weighted by a separate spatial confidence. Thus, the two weighted similarity scores could be combined to produce a more discriminating object association process. By considering these scores as simple probabilities, the most likely combination was to multiply the weighted scores, as seen in equation C.1,

$$\textbf{\textit{Combined score}} = [(C_s)\ x\ S_s]\ x\ [(C_{xy})\ x\ S_{xy}]. \qquad \text{(C.1)}$$

S_s is the spectral similarity score and is weighted by the spectral confidence C_s. Likewise, the spatial similarity score, S_{xy} , is weighted by the spatial confidence score C_{xy}. A maximum combined score would result in a match between the target in the previous frame and a candidate object in the current frame.

In order to match and label detected objects in the current frame, each segmented bounding box is compared against the existing objects from previous frames in a three step process. Figure C.1

provides an overview of these steps, where four moving objects are being tracked from previous frames and four objects have been detected in the current frame.

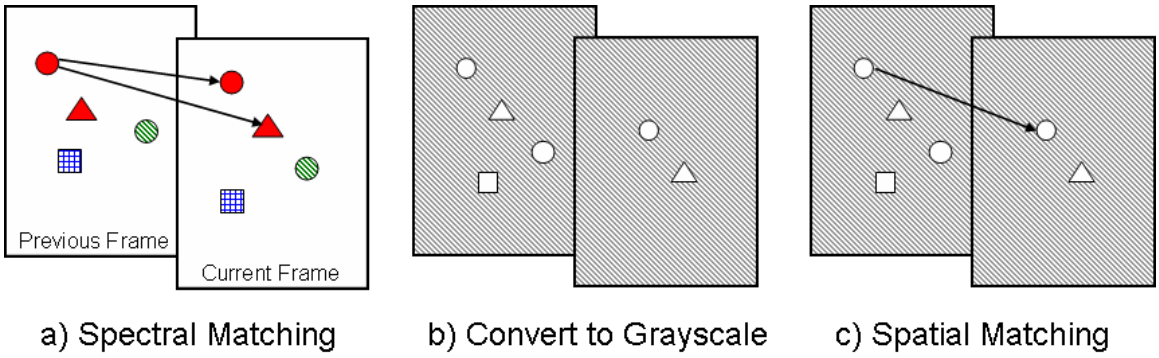


FIGURE C.1 – Three Step Object Association Process.

First (figure C.1, a), each target in the previous frame is represented as a target spectrum and compared against all candidate objects in the current frame using spectral similarity. Considering only the first target track (the red circle) for this example, we can immediately remove the blue square and green circle from the current frame. Second (figure C.1, b), the remaining two candidates are converted to grayscale using the spectral distance from the candidate mean spectrum. Thus, each pixel within a candidate bounding box is given a single value. Similarly, each target is converted to grayscale using the target mean spectrum. Third (figure C.1, c), a spatial similarity score is assigned to each candidate using single-band correlation based on pixel intensity differences. The spectral and spatial similarity scores are then combined to determine the best match for the target. The candidate object with the highest combined similarity to a target is assigned to that track. As a separate system function, track management would handle the cases where there are still multiple matches or where no suitable match is found.

In this simplified example, the first target has multiple spectral matches in the first step—one candidate might be a slightly better match, but both candidates meet the threshold. However, after reducing the data to grayscale in the second step, the spatial comparison in step three results in a better spatial match for the red circle. Significantly, notice that the green circle might very well have been matched to the red circle in a single-band tracker, yet here it was removed in the first step.

C.2 Spectral Similarity Score

The first step in object association is to compare each target to each candidate and compute a spectral similarity score (S_s) via Spectral Angle Mapping (SAM). By processing each existing

target bounding box, or region (R_i), as compared to each candidate region (R_j), a SAM score is computed for each target/candidate pair, as shown in equation C.2,

$$S_s = SAM(Li, Lj) = \cos^{-1}\left[\frac{Li \bullet Lj}{|Li| \cdot |Lj|}\right] \quad . \tag{C.2}$$

L_i is the spectral mean of the ith target and L_j is the spectral mean of the jth candidate. The spectral means are computed through a simple average of the individual pixels in the respective bounding boxes. These pixels could also be weighted either spatially (where center pixels are weighted more heavily than pixels on the edge of the bounding box) or spectrally (where pixels closer to the spectral mean are weighted more heavily). Variations on this technique will need to be evaluated.

Once a similarity score is computed for all target/candidate pairs, a matching threshold is applied and outliers are discarded. At this point, there may be multiple candidate matches for the given target (i.e. a split object) as seen in figure C.2, which may be resolved in the next two object association steps.

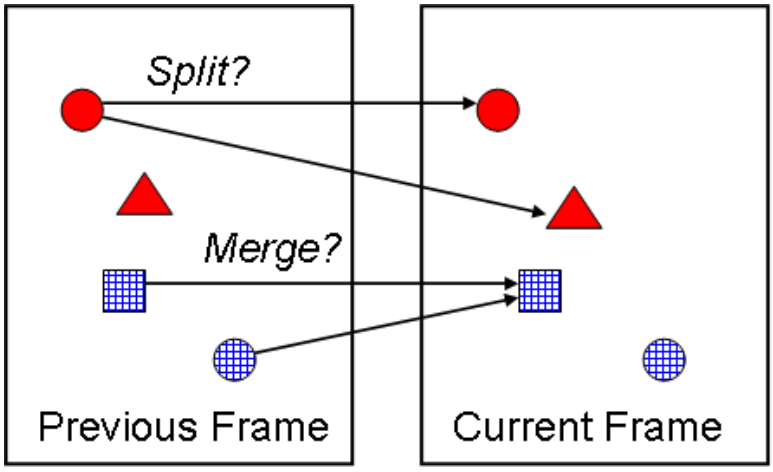


FIGURE C.2 – Multiple spectral matches.

In the top half of this example, the first target (a red circle) matches spectrally with two candidates in the current frame (a circle and triangle, also both red). This could potentially be a single object that has split into multiple objects, or simply two targets with very similar spectra. Note that a single candidate may also be the best spectral match for multiple targets, as seen in the bottom half of figure 21. After spectral matching, SAM scores could be saved as spectral confidence for each target/candidate pair and ambiguous matches that remain after the next two object association steps would be handled as multiple hypotheses as described in Background section.

C.3 Reduce Datasets to Grayscale

The second step in object association is converting the spectral objects into grayscale templates, again using SAM. In this case, all of the pixels in a single candidate bounding box are remapped by assigning the SAM distance to the candidate mean spectrum as the individual pixel values. In similar fashion, all target pixels are remapped using the distance to the target mean spectrum. As seen in figure C.3, three separate objects (either targets or candidates) are segmented by a bounding box.

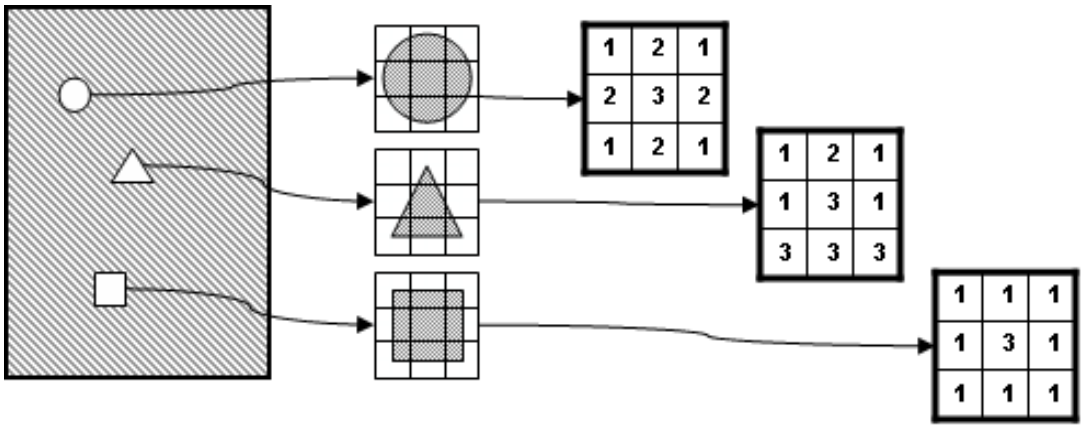


FIGURE C.3 – Convert Spectral Objects to Grayscale.

Each pixel in a bounding box is then given a single value based on the spectral distance of that pixel from the mean spectrum of all the pixels within the bounding box. The notional numbers provided in the figure imply that the shape of the object is preserved in the single-band representation. We now have a set of grayscale targets and set of grayscale candidates—each characterized by a unique spatial distribution of pixel values based on the distance from their respective object mean spectral vectors.

The third and final step in object association is to compute a spatial similarity score by comparing the grayscale targets to the grayscale candidates using the VSAM single-band block-matching method discussed in the background section [Collins:2000]. By comparing a target region (or bounding box) in the previous frame to candidate regions in the current frame, a correlation function *C(d)* is derived, as seen in equation C.3,

$$C(d) = \sum_{x \in R} \frac{W(i,j)\,|I_n(x) - I_{n+1}(x+d)|}{||W||} \qquad \hat{d} = \min_d C(d) \quad . \tag{C.3}$$

To compute the correlation function *C(d)*, we accumulate a weighted sum of absolute intensity differences between each pixel *x* in region *R* and the corresponding pixel *x+d* in the next frame.

The estimated offset distance $\hat{d}$ of the best match is given by the argmin of the correlation function, and the quality of the match is generated from the value of min*C(d)*. The weighting factor *W* is a linear function based on the distance from the center of *R*, giving more weight on the center pixels, as seen in equation C.4. Weighting the center pixels assists in deemphasizing the edge pixels which are more likely to have background content,

$$||W|| = \sum_{x \in R} W(x) \qquad W(x) = \frac{1}{2} + \frac{1}{2}\left(1 - \frac{r(x)}{r_{\max}}\right) . \tag{C.4}$$

The calculations are performed in the *x* and *y* dimensions separately. To better visualize the correlation function, this measure can be considered as a correlation surface over the potential offsets in both *x* and *y*. A correlation surface can be seen in figure C.4 (inverted for easier viewing), where the highest point is actually the minimum value of *C(d)* [Collins:2000].

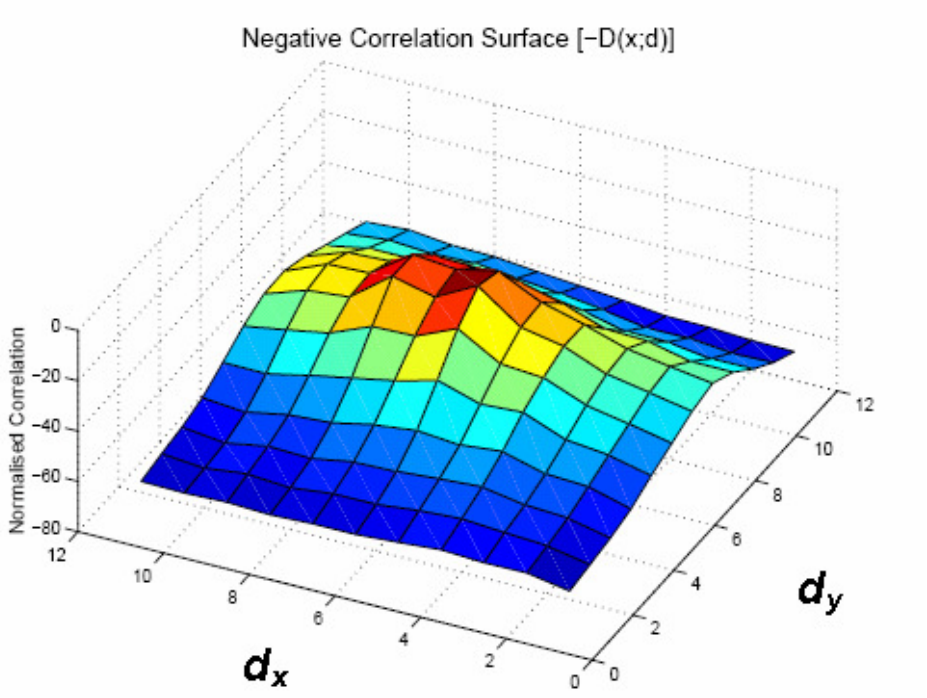


FIGURE C.4 – Correlation Surface [Collins:2000].

After comparing targets to candidates both spectrally and spatially, the combined score is applied to each target/candidate pair. Track management would then assemble the scores and determine which tracks to update in terms of the five possible scenarios (best, stopped, split, merge, new) as described in the Background section.

Works Cited

R. Ackerman. Advanced surveillance spawns new challenges. Signal Magazine, March 2004.

A. Adams. Detecting salient motion, digital video processing final project, rit. Digital Video Processing Final Project, Rochester Institute of Technology, 2006.

C. Anderson, P. Burt, and G. van der Wal. Change detection and tracking using pyramid transformation techniques. In SPIE - Intelligent Robots and Computer Vision, volume 579, page 7278. SPIE, 1985.

J. Barron, D. Fleet, and S. Beauchemin. Performance of optical flow techniques. International Journal of Computer Vision, 12(1):4277, 1994.

B. Bartlett. Improvment of Retrieved Reflectance in the Presence of Clouds. PhD thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2007.

F. Bashir and F. Porikli. Performance evaluation of object detection and tracking systems. In IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS). IEEE, June 2006.

D. Beymer, P.McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In Computer Vision and Pattern Recognition (CVPR), page 495501. IEEE, 1997.

S. Birchfield and S. Rangarajan. Spatiograms versus histograms for region-based tracking. In Computer Vision and Pattern Recognition (CVPR2005), volume 2, pages 1158–1163. IEEE, June 2005.

J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (PETS), pages 125–132, Oct 2003.

S. Blackman and R. Popoli. Design and Analysis of Modern Tracking Systems. Artech

House, Norwood, MA, 1997. ISBN 1-58053-006-0.

S. Brown. DIRSIG User's Manual, Release 4. Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY 14623-5604, 2006. http://dirsig.cis.rit.edu/doc/manual/manual.html.

L. Bruzzone and P. Fernandez. An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. IEEE Transactions on Image Processing, 11(4):452–466, Apr 2002.

L. Bruzzone and S. Serpico. An iterative technique for the detection of land-cover transitions in multitemporal remote-sensing images. IEEE Trans. Geosci. Remote Sensing, 35:858867, July 1997.

G. Byrne, P. Crapper, and K. Mayo. Monitoring land-cover change by principal component analysis of multitemporal landsat data. Remote Sens. Environ., 10:175–184, 1980.

M. Carlotto. Detection and analysis of change in remotely sensed imagery with application to wide area surveillance. IEEE Trans. Image Processing, 6:189202, Jan 1997.

A. Cavallaro, O. Steiger, and T. Ebrahimi. Tracking video objects in cluttered background. IEEE Transactions on Circuits and Systems for Video Technology, 15(4): 575–584, 2005.

R. Collins, A. Lipton, T. Kanade, H. Fujiyoshi, D. Duggins, Y. Tsin, D. Tolliver,

N. Enomoto, and O. Hasegawa. A system for video surveillance and monitoring. Technical Report CMU-RI-TR-00-12, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 2000.

C. . Conaire, N. O'Connor, E. Cooke, and A. Smeaton. Comparison of fusion methods for thermo-visual surveillance tracking. International Conference on Information Fusion (FUSION 2006), 2006a. URL http://www.eeng.dcu.ie/_oconaire/papers/ fusion06/oconaire fusion2006 hyowon-generated.pdf.

C. . Conaire, N. E. O'Connor, and A. Smeaton. Thermo-visual feature fusion for object tracking using multiple spatiogram trackers. Journal of Machine Vision and Applications (in print), 2006b.

J. Engvall, J. Tubbs, and Q. Holmes. Pattern recognition of landsat data based on

temporal trend analysis. Remote Sens. Environ., 6:303–313, 1977.

B. Gnsel, A. M. Tekalp, and P. J. van Beek. Content-based access to video objects: temporal segmentation, visual summarization, and feature extraction. Signal Processing, 66(2):261280, 1998.

R. Gonzalez and R. Woods. Digital Image Processing. Prentice-Hall, Upper Saddle River, NJ, second edition, 2001. ISBN 0-201-18075-8.

I. Haritaoglu, L. S. Davis, and D. Harwood. W4 who? when? where? what? a real time system for detecing and tracking people. FGR98, 1998. http://wiki.cis.rit.edu/bin/view/LIAS/WaspHome. http://www.cis.rit.edu/ lias/wasplite/. Website, 2007.

W. Hu, T. Tan, L. Wang, and S. Maybank. A survey on visual surveillance of object motion and behaviors. IEEE Transactions on Systems, Man and Cybernetics, 34(3): 334–350, August 2004.

E. Ientilucci. Synthetic simulation and modeling of image intensified ccds. Master's thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2000.

G. Jing, C. E. Siong, and D. Rajan. Foreground motion detection by difference-based spatial temporal entropy image. In IEEE Region 10 Conference TENCON, volume A, pages 379–382. IEEE, Nov 2004.

J. Kapur, P. Sahoo, and A. Wong. A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, Image Processing, 29:273285, 1985.

J. G. Kawamura. Automatic recognition of changes in urban development from aerial photographs. IEEE Trans. Syst., Man, Cybern, 1(3), July 1971.

D. Koller, K. Danilidis, and H. Nagel. Model-based object tracking in monocular image sequences of road traffic scenes. Int. J. Comput. Vis., 1993.

L. Latecki and R. Miezianko. Using spatiotemporal blocks to reduce the uncertainty in detecting and tracking moving objects in video. Int. J. Intelligent Systems Technologies and Applications, 1(3-4):376–392, 2006.

L. Latecki, R. Miezianko, and D. Pokrajac. Tracking motion objects in infrared videos. In IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS),

pages 99–104. IEEE, September 2005.

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of DARPA Imaging understanding workshop, pages 121–130, April 1981.

S. McNamara. Using multispectral sensor wasp-lite to analyze harmful algal blooms. Master's thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2007.

T. Meier and K. Ngan. Automatic segmentation of moving objects for video object plane generation. IEEE Trans. Circuits Syst. Video Technol., 8(5):525538, May 1998.

R. Miezianko. Motion Detection and Object Tracking in Grayscale Videos Based on Spatiotemporal Texture Changes. PhD dissertation, CIS Dept., Temple University, Philadelphia, PA, Jan 2006.

R. Miezianko. Personal correspondence with Roland Miezianko. Email and Phone, 2007.

D. of Defense. Dictionary of terms, 2006. http://www.dtic.mil/doctrine/jel/doddict/data/p/04054.html.

J. Parker. Gray level thresholding in badly illuminated images. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 813–819, August 1991.

N. Peterfreund. Robust tracking of position and velocity with kalman snakes. IEEE Trans. Pattern Anal. Machine Intell., 1998.

F. Porikli and O. Tuzel. Object tracking in low-frame-rate video. SPIE Image and Video Communications and Processing, 5685:72–79, March 2005.

K. Price and R. Reddy. Change detection and analysis in multispectral images. Proc. 5th Int. J. Conf. Artificial Intell., pages 619–625, 1977.

D. Reid. An algorithm for tracking multiple targets. IEEE Transactions on Automatic Control, 24(6):843–854, Dec 1979.

P. Shippert. Introduction to hyperspectral image analysis. http://satjournal.tcom.ohiou.edu/pdf/shippert.pdf, 2006. Earth Science Applications Specialist, Research Systems, Inc. Website Tutorial.

A. Singh. Digital change detection techniques using remotely sensed data. Int. J. Remote Sensing, 10(6):9891003, 1989.

C. Stauffer and W. Grimson. Adaptive background mixture models for real-time track-

ing. In Computer Vision and Pattern Recognition 1999 (CVPR ’99), volume 2, page 2246. IEEE, June 1999.

H. Tao, H. Sawhney, and R. Kumar. Object tracking with bayesian estimation of dynamic layer representation. IEEE Trans. Pattern Anal. Machine Intell., 24(1):7589, Jan 2002.

J. Taylor. Introduction to Error Analysis. University Science Books, Sausalito, CA, second edition, 1997. ISBN 0-935702-75-x.

A. M. Tekalp. Digital Video Processing. Prentice-Hall Signal Processing Series, Upper Saddle River, NJ, 1995. ISBN 0-13-190075-7.

Y.-L. Tian and A. Hampapur. Robust salient motion detection with complex background for real-time video surveillance. In IEEE Workshop on Motion and Video Computing (WACV/MOTION’05), volume 2, pages 30–35. IEEE, 2005.

J. Townshend, C. Justice, and C. Gurney. The impact of misregistration on change detection. IEEE Trans. Geosci. Remote Sensing, 30:10541060, Sept 1992.

L.Wixson. Detecting salient motion by accumulating directionally-consistent flow. IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(8):774–780, Aug 2000.

C. Wren, A. Azarbayejani, T. Darrell, and A. Pentland. Pfinder: Real-time tracking of the human body. IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):780785, 1997.

R. Yuhas, A. Goetz, and J. Boardman. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm. In Summaries of the Third Annual JPL Airborne Geoscience Workshop: AVIRIS, volume 1, pages 147–149. JPL, 1992. SEE N94-16666 03-42.

S. Zhou, R. Chellappa, and B. Moghaddam. Appearance tracking using adaptive models in a particle filter. In In Proc. of 6th Asian Conference on Computer Vision (ACCV), Jan 2004.