CHANGE DETECTION METHODS
FOR HYPERSPECTRAL IMAGERY

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering

By

KARMON MARIE VONGSY
B.S., Wright State University, 2005

2007
Wright State University
WRIGHT STATE UNIVERSITY

_________________________________________
Arnab K. Shaw, Ph.D.
Thesis Director

_________________________________________
Fred Garber, Ph.D.
Department Chair

Committee on Final Examination

_________________________________________
Arnab K. Shaw, Ph.D.

_________________________________________
Brian Rigling, Ph.D.

_________________________________________
Kefu Xue, Ph.D.

_________________________________________
Joseph F. Thomas, Jr., Ph.D.
Dean of the School of Graduate Studies
ABSTRACT


This thesis studies the detection of changes using hyperspectral images. Change Detection (CD) is the process of identifying and examining temporal and spectral changes in signals. Detection and analysis of change provide valuable information of possible transformations in a scene. Hyperspectral imaging (HSI) sensors are capable of collecting data at hundreds of narrow spectral bands. Such sensors provide high-resolution spatial and spectrally rich information that can be exploited for CD. This work develops and implements various CD algorithms for detection of changes using Hyperspectral images. The main objectives are to study and develop different HSI change detection algorithms. The explored methods were implemented in order to compare the performance on close-in HSI data. The methods studied in this thesis include, Image Differencing, Image Ratioing, Principal Component Analysis, Linear Chronochrome, a modified Correlation Coefficient and a Kernel Dissimilarity Measure. Hyperspectral imagery of different scenarios was collected and used to test and validate the methods presented in this study. The algorithms were implemented using MATLAB, and the performance of algorithms is presented in terms of false alarm rates and missed changes.
# TABLE OF CONTENTS

1. **INTRODUCTION** ......................................................................................................................... 1

   1.1. Thesis Motivations .................................................................................................................. 1

   1.2. Thesis Objectives .................................................................................................................... 2

   1.3. Literature Survey ................................................................................................................... 2

   1.4. Thesis Organization ................................................................................................................. 6

2. **BACKGROUND INFORMATION** ............................................................................................ 8

   2.1. Change Detection ................................................................................................................... 8

   2.2. Hyperspectral Imaging .......................................................................................................... 11

3. **CHANGE DETECTION (CD) ALGORITHMS** ................................................................... 17

   3.1. Image Differencing ................................................................................................................. 17

   3.2. Image Ratios .......................................................................................................................... 18

   3.3. Principal Component analysis (PCA) ...................................................................................... 18

   3.4. Linear Chronochrome Change Detection .............................................................................. 22

   3.5. Maximum Likelihood Correlation Coefficient ..................................................................... 25

   3.6. Kernel Change Detection Dissimilarity Measure ................................................................. 28

4. **HSI CAMERA AND DATA** ............................................................................................. 40

   4.1. Hyperspectral Sensor .......................................................................................................... 40

   4.2. HSI Data Collection .......................................................................................................... 48

   4.3. Spectral Signature Evaluation ............................................................................................ 49

5. **CHANGE DETECTION SIMULATION RESULTS** .......................................................... 52
5.1. Change Detection using Image Differencing ......................................................... 52
5.2. Change Detection using Image Ratioing ............................................................... 55
5.3. Change Detection using PCA ............................................................................. 57
5.4. Linear Chronochrome Change Detection Results ............................................... 61
5.5. Maximum-Likelihood Change Detection Results ................................................ 63
5.6. Change Detection using Kernel Approach ........................................................... 68
5.7. Inter-band Results using Correlation Coefficient ................................................ 72
5.8. Time Lapse Results on Grass Data .................................................................... 76

6. CONCLUSIONS AND FUTURE WORK ................................................................ 84

6.1. Concluding Summary ......................................................................................... 84
6.2. Future Work ........................................................................................................ 86

APPENDIX A ........................................................................................................... 87

A.1 Derivation of the maximum likelihood estimate ............................................... 87
A.2 Derivation of the kernel dissimilarity measure ................................................... 89
A.3 Mercer’s Theorem .............................................................................................. 91

APPENDIX B ........................................................................................................... 92

B.1 MATLAB code for implementation of Principal Component Analysis ............. 92
B.2 MATLAB code for the Linear Chronochrome Change detection ....................... 94
B.3 MATLAB code for implementation of Modified Correlation Coefficient ........... 96
B.4 MATLAB code for implementation of the kernel dissimilarity measure .......... 98
B.5 MATLAB code ROC curve generation for multiple time lapse grass experiment

................................................................. 101

REFERENCES ................................................. 103
LIST OF FIGURES

FIGURE 2.1-1 CHANGE DETECTION PROCESS .................................................................10
FIGURE 2.2-1 SPECTRAL ANALYSIS OF HSI CUBE ......................................................12
FIGURE 2.2-2 HSI DATA CUBE .................................................................................13
FIGURE 2.2-3 CUBE-TO-CUBE CHANGE PROCESS ..................................................15
FIGURE 2.2-4 BAND-TO-BAND CHANGE PROCESS ..................................................16
FIGURE 3.3-1 CHANGE DETECTION PRINCIPLE COMPONENT ANALYSIS ............19
FIGURE 3.3-2 PCA ALGORITHM FOR CHANGE DETECTION (CUBE-TO-CUBE) .......21
FIGURE 3.4-1 CHRONOCHROME ALGORITHM (CUBE-TO-CUBE) .........................24
FIGURE 3.5-1 CORRELATION COEFFICIENT ALGORITHM (BAND-TO-BAND) .......27
FIGURE 3.6-1 MAPPING NONLINEAR DATA .............................................................29
FIGURE 3.6-2 XOR EXAMPLE IN INPUT SPACE .........................................................32
FIGURE 3.6-3 ENCIRCLED DATA KERNEL EXAMPLE ...............................................33
FIGURE 3.6-4 KERNEL MATLAB CODE .................................................................35
FIGURE 3.6-5 KERNEL FEATURE SPACE GEOMETRY ...............................................36
FIGURE 3.6-6 KERNEL CHANGE DETECTION FLOW CHART ...................................37
FIGURE 3.6-7 DISSIMILARITY MEASURE GEOMETRY IN FEATURE SPACE ...............38
FIGURE 4.1-1 PANTERA TF 1M60 ............................................................................41
FIGURE 4.1-2 CAMERA OPTICAL PATH ..................................................................42
FIGURE 4.1-3 PANTERA TF 1M60 RESPONSIVITY ..................................................43
FIGURE 4.1-4 TRANSMITTANCE OF LCTF .............................................................44
FIGURE 4.1-5 VISIBLE SPECTRUM (WWW.YORKU.CA/EYE/SPECTRU.HM) ...........45
FIGURE 4.1-6 INITIALIZATION DIALOG BOX ..........................................................47
FIGURE 4.1-7 HSI DATA ..........................................................................................49
FIGURE 4.3-1 GRASS DATA IMAGE STATISTICS .......................................................50
FIGURE 5.8-6 ROC CURVES FOR GRASS DIFFERENCE RESULTS.................................................83
LIST OF TABLES

TABLE 2-1 CHANGE DETECTION MODES.................................................................14
TABLE 3-1 XOR TRUTH TABLE ............................................................................30
TABLE 5-1 RESULTS OF CUBE-TO-CUBE SIMULATION.......................................71
TABLE 5-2 SUMMARY OF GRASS DATA ..............................................................77
TABLE 6-1 COMPUTATION TIME OF CHANGE DETECTION ALGORITHMS ...........85
LIST OF EQUATIONS

EQUATION 2-1...........................................................................................................9
EQUATION 3-1.............................................................................................................17
EQUATION 3-2.............................................................................................................18
EQUATION 3-3.............................................................................................................20
EQUATION 3-4.............................................................................................................22
EQUATION 3-5.............................................................................................................22
EQUATION 3-6.............................................................................................................22
EQUATION 3-7.............................................................................................................22
EQUATION 3-8.............................................................................................................22
EQUATION 3-9.............................................................................................................23
EQUATION 3-10..........................................................................................................23
EQUATION 3-11..........................................................................................................23
EQUATION 3-12..........................................................................................................25
EQUATION 3-13..........................................................................................................25
EQUATION 3-14..........................................................................................................25
EQUATION 3-15..........................................................................................................25
EQUATION 3-16..........................................................................................................25
EQUATION 3-17..........................................................................................................26
EQUATION 3-18..........................................................................................................26
EQUATION 3-19..........................................................................................................26
EQUATION 3-20..........................................................................................................30
EQUATION 3-21..........................................................................................................34
EQUATION 3-22..........................................................................................................34
EQUATION 3-23..........................................................................................................39
ACKNOWLEDGMENTS

I would like to sincerely thank Dr. Arnab Shaw and Dr. Devert Wicker for all of their time, effort, insight and assistance during my research and academic journey. I would also like to thank Dr. Kefu Xue and Dr. Brian Rigling for their time and input while serving on my committee. I extend special thanks for their assistance, suggestions, and friendly counseling as I worked through problems, both in my research and coursework.

I would like to thank Henry Griffith and Andrew Kondrath for all their encouragement, wisdom and assistance. I thank them for their continued friendship and guidance. I would like to express my gratitude to Maj. Michael J. Mendenhall, Ph.D. for his detailed and constructive comments and excellent advice during the preparation of this thesis. I am grateful for Dr. James Olson, Ph.D. and Maj. Robert Spence, M.D. for providing their knowledge and editing assistance.

Last but not least, I would like to thank my entire family. My husband and children have been my motivators. Their infinity of love and patience has allowed confidence and encouragement on my journey of knowledge. Finally, I thank my parents for instilling in me this devotion and appreciation of learning.
1. INTRODUCTION

1.1. Thesis Motivations

The ability to detect and characterize changes in images of the same scene taken at different times has been of interest to image processing researchers for many years due in large part to the immense number of applications in diverse disciplines [1]. Typical EO/RGB imagery has limited capability for detecting slight change resulting in a high false alarm rate. Hyperspectral imagery (HSI), with its richness of detail spectral information has greater detection capabilities. Change in HSI data at different passes may be due to object motion or due to change in spectrum. Motion is mostly defined as a variation referenced to location, whereas spectral change is due to change in object status. Military and law enforcement have interest in surveillance and tracking of dismounts and vehicles. Change detection of vehicle or dismount movement could benefit surveillance by initiating the operator of intruders or activity. Change detection tracking can aid in the rapid location, identification and tracking of dismounts or equipment that pose a potential threat. Detection of change involves quickly assessing the impact of or damage from an event or action. Hyperspectral imaging offers a relatively new and emerging technology to achieve this goal. Thereby the goal of this thesis is to develop and implement algorithms for dismount detection and tracking with few false alarms and missed detections using hyperspectral images. The challenge is the high-dimensionality and complexity of HSI data.
1.2. Thesis Objectives

The following will be achieved:

- Provide background in the field of CD and HSI
- Explore and compare existing CD methods
- Exploit the spectral richness of HSI data
- Implement several CD algorithms on HSI data and compare their performance using change data
- Investigate feasibility of HSI-based CD using different methods
- Discuss results and future research directions

1.3. Literature Survey

To ensure that the relevant and most applicable processing techniques were utilized, brief surveys of the literature encountered for existing CD methods and provide an introduction in the CD research field. This literature survey demonstrates the importance of CD analysis for many diverse applications.

Many documented techniques are available for multi-temporal data sets. These are primarily set in remote sensing applications [2, 3, 4]. Coppin and Bauer [3] provide comprehensive summaries of CD methods and techniques. They review the methods and the results of CD documented in literature up to 1995. This work was an extension of previous publications by Singh [4] who provided the first comprehensive summary of methods and techniques of CD. Coppin and Bauer conclude that CD is a difficult task to perform accurately. Jensen [2] dedicated a chapter in *Introductory Digital Image*
Processing which presented nine CD algorithms with diagrams. He emphasized the importance of selecting the appropriate CD algorithm for each application and explained how implementation could profoundly affect the qualitative and quantitative estimates of the change. The optimal selection of the CD algorithm remains a popular topic in the literature. Much of the current research on CD was applied to multispectral imagery associated with remote sensing. Though the literature positions CD as a key component for remote sensing images; CD algorithms can generally be applied to most image data (such as, in this study, hyperspectral images.) Overall, the CD process does not deviate from the sensor data, no matter if the source is ground-based, airborne, or satellite images.

Singh [4] identified image differencing as the most widely applied CD algorithm. According to Coppin and Bauer [3], image differencing generally performs better than other methods of CD. They suggested a standardization of the differencing algorithm to minimize the occurrence of identical change values showing different change events and introduced a method by which they divided the difference by the sum. Jensen [2] reported that while no change pixels or minor radiance values are distributed around the mean, pixels of significant changes are distributed in tails of the distribution.

Similar to image differencing techniques, image ratios were developed to enhance spectral differences. Image ratios use the ratio (instead of a difference) of the pixel intensities of the two images to portray the change between two images [4]. Weydahl [5] noted that image ratios are less sensitive than differencing to multiplicative noise in synthetic aperture radar (SAR) imagery compared with image differences. Smits and Annoni [6] reported a CD methodology that was based on explicit user requirements in
terms of example imagery. They described the likelihood of false alarm as well as misclassification probabilities. In general, pixel-wise CD methods often suffer from false alarm rates that may be prohibitive for certain applications [6].

In the remote-sensing community, it is common to utilize linear transformation for CD analysis. A number of image analysis approaches to CD are linear techniques, meaning a change at each image location is associated with some linear transformation [7]. Collins and Woodcock [7] state the best known linear CD technique is multidata principal component analysis (PCA). Wiemker et al. [8] presented the PCA as a linear transformation which defines a new, orthogonal coordinate system such that the data can be represented without correlation. They considered a bitemporal feature space for a single spectral band which resulted in a linear relation between unchanged pixels lying in an elongated cluster along a principal axis. Alternately, one can apply PCA to difference images as in Gong [9], where the first one or two principal component images are assumed to represent changed regions. Collins and Woodcock [7] presented a linear technique which is essentially a multitemporal generalization of the well-known Kauth-Thomas transformation. This transformation orthogonalizes spectral vectors taken directly from a bitemporal image and produces three stable components plus a change component associated with inter-date differences. Multiple authors indicate that PCA based CD does not perform as well as other simpler techniques [4]. Here, we will implement PCA as one of several methods to detect change in hyperspectral imagery.

CD can be accomplished by performing statistical analysis to determine spectral differences. Schaum and Stocker [10] described a CD algorithm called “chronochrome” which assumed linear change between passes. Chronochrome detection estimates
Correlation analysis can provide basic understanding about how different variables are related. Nielsen et al. [11] introduced the Multivariate Alteration Detection (MAD) procedure, an application of a statistical transformation referred to as canonical correlation. This procedure determines coefficients so that the positive correlation between two images is minimized. Cooley and Lohnes [12] expanded on the multivariate statistical canonical correlation analysis technique. Woo et al. [13] performed earthquake damage via a CD algorithm based on the sample correlation coefficient for a windowed area. Koeln and Bissonnette [14] introduced a cross-correlation analysis measure which displayed the differences between a land cover database and a recent multispectral image. Eichel et al. [15] introduced a maximum likelihood estimator for SAR imagery for CD. Based on this research, we will employ the maximum likelihood estimator by disregarding the phase information in augmenting the coefficient analysis to represent HSI.

Support Vector Machine (SVM) is based on statistical learning theory and has the goal of determining the location of decision boundaries that produce the optimal separation of classes [16]. Vapnik [16] formulated a linear classification problem in feature space using kernel functions. Camps-Valls et al. [17] described and outlined kernel methods, such as SVM, and noted the demonstrated excellent performance in hyperspectral data classification through accuracy and robustness. Nemmour and Chibani [18] presented SVM-based CD and experimentation on land cover change using multispectral images. Desobry et al. [19] describe a kernel CD algorithm based on single-class SVMs. The change is expressed as a dissimilarity measure based on the arc distance in
feature space. This CD algorithm offers the potential for optimization. For this reason, the kernel dissimilarity measure was researched and implemented in this study.

1.4. Thesis Organization

In Chapter 2, a presentation of the general concepts about CD and HSI is provided. A description of hyperspectral data is supplied.

In Chapter 3, a discussion of current CD algorithms is described involving imagery data. In addition, this chapter shows how the maximum likelihood correlation coefficient is used to analyze hyperspectral data. It also offers the methodology to achieve kernelization CD for HSI data.

In Chapter 4, an introduction to the hyperspectral sensor and an explanation of the acquired hyperspectral is explained with the collections scenarios.

In Chapter 5, an explanation of the algorithm implementation is provided with the visual results obtained with each technique.

In Chapter 6, details of the conclusions from each algorithm performance is provided with additional areas for future enhancements.

Appendix A details Mercer’s Theorem and derivations of algorithms for this study.
Appendix B shows the MATLAB code used to implement the various CD methods utilized in this work.
2. BACKGROUND INFORMATION

2.1. Change Detection

Change processing on imagery data involves the detection of a set of pixels that have undergone a significant change relative in a previous data sequence. This change in time is typically referred to as temporal change and is performed as a systematic CD study involving two sets of data prepared at different times [20]. The changes may be due to object movement, insertion, deletion, removal or deformation, and the changes are usually affect the spectral signatures at same pixel locations of two sets of images of the same scene.

Before analyzing CD it is essential the two sets of data are accurately registered. Varshney [20] noted that registration accuracy of less than one-fifth of a pixel is required to achieve a CD error rate of less than 10%. The underlining fundamental assumption when applying any CD algorithm is that when there is a difference in spectral response of a pixel between images of two time lapse informatics, a change is detected. CD analysis typically generates a correspondence image from an image pair showing any changes. (Figure 2.1-1) Usually, in the comparison process, two corresponding pixels belonging to the same location in an image pair are determined on the basis of a quantitative measure.
If this measure exceeds a predefined threshold a change is labeled. A binary image, $B$, identifies the changed region, $[1]$, where

$$B(x) = \begin{cases} 
1, & \text{if pixel corresponds a significant change} \\
0, & \text{elsewhere} 
\end{cases}$$

Equation 2-1

Two images or set of images is the minimum requirement to identify change however the pair can be a successive series of images.
Figure 2.1-1 Change detection process
2.2. Hyperspectral Imaging

The basic notation for hyperspectral imagery in the context of change detection is presented here. A hyperspectral cube can be thought of as a stack of images, one on top of the other, where each image in the stack represents the instantaneous field of view (IFOV) at a particular wavelength-band of light. Thus a two-dimensional image in view is represented by a three-dimensional image-cube where the third dimension represents optical wavelength. Each point or pixel in the image collected by a hyperspectral sensor contains information on the entire spectrum of light, which theoretically can be continuous but it is usually discrete. However, the discrete HSI spectrum contains considerably higher resolution data than normal electro-optics (EO) or RGB images [21]. Different materials have different wavelength-dependent absorption. These hyperspectral images of reflected energy are known as spectral signatures. Figure 2.2-1 shows how hyperspectral imaging spectrometers produce a complete spectrum for every pixel of the image (Figure 2.2-1).
Figure 2.2-1  Spectral analysis of HSI cube
The format of a typical HSI cube is depicted in Figure 2.2-2. It describes the pixel vector as, \( p_n = (x_n, y_n) = [x_1, y_1, \ldots, x_N, y_N]^T \), with each component \((x_1, y_1)\) representing a pixel location in a frame image, \(\lambda_N\) in a specific spectral range [22]. At each pixel location an intensity value as a function of wavelength is recorded which provides the continuous spectral signature. In this cube, the x and y axis specify the spatial dimensions of the image, whereas the depth axis or \(\lambda\) denotes the number of bands in the hyperspectral data. As an image matrix this is a three-dimensional array of pixel intensities.

![Figure 2.2-2 HSI data cube](image)

Although HSI sensors act as still-cameras, the images collected at contiguous wavelengths can be considered as video frames. The collection of a 3D HSI image cube may take 100s of milliseconds to a few seconds. Object motion during HSI image collection results in movement of the objects through different HSI spectral bands. In fact, motion video can be created by displaying the images collected at successive
spectral bands. More interestingly, motion tracking can be implemented by applying CD to the images at successive HSI bands. An exploration of the use of HSI cubes for CD is capable in two types of modes: cube-to-cube CD that looks for changes in two image cubes collected in two passes and CD between spectral bands within a single cube in one pass. All of the CD methods discussed can be used in either of these two modes (Table 2-1).

<table>
<thead>
<tr>
<th>Mode</th>
<th>Description</th>
<th>Use</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cube-to-Cube</td>
<td>CD frequency band to same frequency band analysis of different cubes</td>
<td>Most common form of CD process</td>
<td>Detects temporal change</td>
</tr>
<tr>
<td>Band-to-Band</td>
<td>CD within same cube of successive frequency bands analysis</td>
<td>Allows CD process by frequencies</td>
<td>Depends on capabilities of sensor</td>
</tr>
</tbody>
</table>

Table 2-1 Change detection modes

These modes are demonstrated in the figures below which show HSI cube data.
Figure 2.2-3  Cube-to-cube change process
Figure 2.2-4 Band-to-Band change process
3. CHANGE DETECTION (CD) ALGORITHMS

3.1. Image Differencing

Image differencing is one of the simplest and useful CD techniques. Differencing attempts to find the areas in a scene which are changing and it usually corresponds to a movement or motion in the scene. Image differencing involves subtracting of the intensity values at same pixel locations of two images collected at two different periods of time. The two co-registered images are compared pixel-by-pixel and pixels associated with changed areas produce values significantly different from those pixels associated to unchanged areas. The subtraction results in positive and negative values in areas of change and zero values in areas of no change in a new third image.

\[
B_{xy} = \begin{cases} 
1 & \text{if } |I_x - I_{x+1}| > \text{threshold} \\
0 & \text{otherwise}
\end{cases}
\]  

Equation 3-1

The third image or difference image is analyzed to obtain a change or no change classification by applying a threshold. The decision rule is the most critical step of any CD method. Only the pixels in the difference image above the threshold will correspond to a change at that location. After thresholding, a difference binary image \(B_{xy}\) is obtained, all pixels in which show change with a value 1 (white) and the pixels with no change have a value of 0 (black).
3.2. Image Ratios

Image ratios or band ratios involve the same logic, except a ratio is computed and the pixels that did not change have a ratio value near 1 in the ratio image. This process divides brightness values of pixels in one band by the brightness values of their corresponding pixels in another band to create the output image. These ratios may enhance or subdue certain attributes found in the image, depending on the spectral characteristics in each of the two bands or images.

\[
\text{Image Ratioing} = a \frac{I_\lambda}{I_{\lambda+1}}
\]

Equation 3-2

In the equation above the parameter, \( a \) represents a possible scaling factor which can vary depending on the application. The ratio binary image is the new image of data created by the division of a set of bands for each pixel after a decision rule is introduced. Since values close to 1 correspond to no change, the algorithm must establish a multilevel threshold to determine where a change occurs. The operator should be aware of the possibility of division by zeros [2]. This method has an important advantage in that ratios minimize the variations in illumination.

Both image differencing and image ratios are considered to be image algebra CD techniques because their basis is on mathematical manipulation.

3.3. Principal Component analysis (PCA)

Principal Component Analysis is a linear transformation technique and probably the most common of these techniques. The main principal of the PCA approach is to use as input a set of images and to reorganize them via a linear transformation, such that the
output images are linearly independent. The new coordinate system for the data is projected such that the greatest variance lies on the first axis or the first principal component and the second greatest variance on the second axis. This technique is usually used to reduce the number of spectral bands or in compression schemes. In CD studies, the consequence of this linearization is that unchanged pixels or common information shared by a pair of images are expected to lie in a narrow elongated cluster along a principal axis equivalent to the first component (PC1). On the contrary, pixels containing a change would be more unique in their spectral appearance and would be expected to lie far away from this axis (PC2). Refer to Figure 3.3-1 [8].

Figure 3.3-1  Change detection principle component analysis
Thereby the magnitude of change is quantified by the magnitude of the second principal component (PC2) given as $c_\lambda = e^{T}_{\lambda,2}(p_{\lambda}(::) - m_{\lambda})$ where $e^{T}_{\lambda,2}$ is the second eigenvector of the overall zero-mean covariance matrix (2 X 2),

$$
C = \begin{bmatrix}
\sigma_1^2 & \rho_{12}\sigma_1\sigma_2 \\
\rho_{12}\sigma_1\sigma_2 & \sigma_2^2
\end{bmatrix}.
$$

Equation 3-3

The algorithm steps are illustrated in the figure 3.3-2. Note the illustration is depicted for cube-to-cube CD analysis but can easily be modified for an inter-band solo cube.

Thus the use of PCA for CD involves examining the least-correlated components as represented by the 2nd principal component, which reflects the changes in the data under investigation.
Figure 3.3-2 PCA algorithm for change detection (cube-to-cube)

Vectorize Data Cubes

Im1 = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix}

Im2 = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_N \end{bmatrix}

Data = \begin{bmatrix} X_1 & Y_1 \\ X_2 & Y_2 \\ \vdots & \vdots \\ X_N & Y_N \end{bmatrix}

c_y = E[(\bar{x}_i - \bar{\mu}_i) \bar{x}_j - \bar{\mu}_j)]]

Perform Eigen Analysis and associated Principal Components

Reshape Data

Change Information

PC1

PC2
3.4. Linear Chronochrome Change Detection

This algorithm proposed by Schaum and Stoker [10] is based on multivariate background statistics and utilizes a linear predictor. Theiler and Perkins [23] analyzed and implemented chronochrome analysis during their research on anomalous CD. The chronochrome algorithm estimates the background in the test image (data set x) as a linear function of the reference image (data set y), and detects changes on the difference image. A Gaussian assumption is then made on the collected data sets with following covariance matrices,

\[
X = E\{(x - \mu_x)(x - \mu_x)^T\}, \text{ where } \mu_x = E\{x\} \tag{3-4}
\]

\[
Y = E\{(y - \mu_y)(y - \mu_y)^T\}, \text{ where } \mu_y = E\{y\}. \tag{3-5}
\]

Before employing the change detection algorithm, zero mean images are formed for both image data frames. The average pixel for each pixel is subtracted to produce an image with zero mean. The centered covariance and cross-covariance is computed before fitting a linear estimate to y-data as a function of the x-data.

\[
X = \langle xx^T \rangle
\]

\[
Y = \langle yy^T \rangle \tag{3-6}
\]

\[
C = \langle yy^T \rangle
\]

A linear estimate is fit to the y-data as a linear function of the x-data.

\[
y = Lx \tag{3-7}
\]

\(L\) is the optimal vector Weiner Filter solution to minimizing \(<(y - Lx)^2>\) w.r.t. \(L\). and it is given by,

\[
L = CX^{-1}. \tag{3-8}
\]
Accordingly, the linear predictor is derived as,
\[ y = CX^{-1}x. \] \hspace{1cm} \text{Equation 3-9}

Proceeding to conduct CD between \( y \) and \( y = Lx \) produces the error image signal
\[ \varepsilon_{cc} = (y - CX^{-1}x) \] \hspace{1cm} \text{Equation 3-10}

The largest changes \( \varepsilon_{cc} \), in the Mahalanobis distance sense is given by the largest values of Equation 3-8. Thresholding the output of such a detector represents the final step of a fully anomalous method for spectral/temporal CD. [23]

\[ \varepsilon_{cc}^T E_{cc}^{-1} \varepsilon_{cc} > \text{threshold} \] \hspace{1cm} \text{Equation 3-11}

where,
\[ E_{cc} = \langle \varepsilon_{cc} \varepsilon_{cc}^T \rangle = Y - CX^{-1}C^T \]

A chronochrome CD algorithm diagram for a cube-to-cube analysis is shown in Figure 3.4-1.
Figure 3.4-1 Chronochrome algorithm (cube-to-cube mode)

- Calculate the error image between consecutive frames.
- Update the estimated parameters.
- Repeat the process.

Mathematically:

\[ e_{c} = (y - CX)^{\top} \]

Where:

- \( e \) is the error image.
- \( y \) is the actual image.
- \( C \) is the estimated parameters.

Example:

\[ X = E[ x - (x - y)^{\top} ] \]

Other values can be calculated similarly.
3.5. Maximum Likelihood Correlation Coefficient

This approach uses the maximum likelihood method to estimate the change parameter [15] incorporated in the joint conditional probability density function (PDF) of an image pair. The HSI cubes or frames are converted into long vectors and the measured reflectivity at pixel position \( n \) in two vector images are denoted by \( x_n \) and \( y_n \). These measured values are represented by the models,

\[
x_n = s_n + n_1
\]

\[
y_n = \alpha s_n + (\sqrt{1 - \alpha^2}) z_n + n_2
\]

where, \( s_n \) denotes the true reflectivity and \( n_1 \) and \( n_2 \) are additive noise terms. The amount of change between the first image, \( x_n \) and the second, \( y_n \) is given by \( \alpha \) which ranges from 0 to 1. The random variable \( z_n \) is completely uncorrelated with \( s_n \) and it represents the change that has occurred. In vector notation, the observation can be written as,

\[
\eta = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} s_n \\ \alpha \end{bmatrix} + \begin{bmatrix} 1 \\ \sqrt{1 - \alpha^2} \end{bmatrix} z_n + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}
\]

Under the assumption that the random variables have zero-mean, are Gaussian and are mutually independent the conditional probability density function (PDF) follows as [14]:

\[
P(\eta | \alpha) = \frac{1}{\pi^2 |Q|} \exp(\eta^T Q^{-1} \eta)
\]

where,

\[
Q = E\{\eta \eta^T\} = \begin{bmatrix} \sigma_s^2 + \sigma_n^2 & \alpha \sigma_s^2 \\ \alpha \sigma_s^2 & \sigma_z^2 + \sigma_n^2 \end{bmatrix}
\]
The maximum likelihood estimate is the value of $\alpha$ that maximizes equation 3-7. This can be computed by taking the logarithm and deleting any terms that are not a function of $\alpha$. See the derivation in appendix A.

$$L = -cN \ln\left[\hat{\sigma}_s^4 \left(1 - \alpha^2\right) + 2\sigma_n^2\hat{\sigma}_s^2 + \sigma_n^4\right] - \sum_{n=1}^{N} \left[ \frac{(x_n^2 - 2x_ny_n\alpha + y_n^2)}{(1 - \alpha^2)\hat{\sigma}_s^2} \right]$$ \hspace{1cm} \text{Equation 3-17}

The next step in finding $\alpha$ that maximizes the above function is to differentiate this expression with respect to $\alpha$ and set to zero

$$\frac{\partial L}{\partial \alpha} = 2N\alpha\hat{\sigma}_s^4 + \frac{2x_ny_n}{(1 - \alpha^2)\hat{\sigma}_s^2} + \frac{2(x_n^2 - 2x_ny_n\alpha + y_n^2)\alpha}{(1 - \alpha^2)^2\hat{\sigma}_s^2} = 0$$ \hspace{1cm} \text{Equation 3-18}

Solving the likelihood equation above for $\alpha$ yields the maximum likelihood coefficient used in CD. The Maximum likelihood estimator of $\alpha$ is given by [15],

$$\alpha_{ML} = \frac{2\sum_n x_n y_n}{\sum_n x_n^2 + \sum_n y_n^2}$$ \hspace{1cm} \text{Equation 3-19}

Figure 3.5-1 illustrates the correlation coefficient algorithm for the single-pass solo cube scenario.
Figure 3.5-1 Correlation coefficient algorithm (band-to-band)

Calculate Correlations between each adjacent frame

\[
\alpha_{ML} = \frac{2 \sum_{n} x_n y_n}{\sum_{n} x_n^2 + \sum_{n} y_n^2}
\]

3 x 3 sliding window performs vectorization on consecutive frames

Change Information
3.6. Kernel Change Detection Dissimilarity Measure

This section explains a CD process based on kernelization techniques [19]. The comparison of two decision regions at different points in time to achieve CD is necessary. However, this can result in nonlinear data. Using kernels, a mapping of the nonlinear input space to a feature space generates linearly separable data. This description implies that a planar separator in the high-dimensional space of feature vectors is a curved separator in the low-dimensional space of the raw input variables. Figure 3.6-1 depicts how a kernel can map nonlinear data to a feature space where it is linearly separable by a hyperplane.
Figure 3.6-1  Mapping nonlinear data
Kernel functions work by embedding of data in a higher-dimensional space supplied by the kernel trick. The most common kernels are linear, polynomial, radius basis function and sigmoidal [24]. A kernel can be mathematically described as a dot product between projections of input variables,

$$ K(x, y) = \Phi(x) \cdot \Phi(y) $$ \hspace{1cm} \textbf{Equation 3-20}

A kernel function defined as an inner product satisfies the Mercer Theorem (see Appendix A). The technique to replace $\Phi(x) \cdot \Phi(y)$ by $K(x, y)$ was described by Vapnik [15] and is called the kernel trick. The trick is used in support vector machines or classifiers [22] when the classes are not linearly separable and allow the data to be mapped to another feature space where the data is linearly separable. This can be accomplished with no extra computation cost except for the cost of the computation of the kernel function $K(x, y)$. The feature map is never explicitly computed.

To demonstrate the kernel idea a consideration of two simple examples is illustrated in MATLAB below. The first test example is the exclusive-or (XOR) problem consisting of four inseparable points in 2-dimensional input space. The input space in Figure 3.6-2 is a scatter plot of the XOR function with the corresponding truth table.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A $\cup$ B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

\textbf{Table 3-1 XOR truth table}

The input was then translated into feature space via a polynomial kernel and expanded into 3-dimensions through a linear fit hyperplane. Figure 3.6-3 is an example of a bulls-
eye pattern show a nonlinear boundary between two sets of data. After mapping the data it is expanded into feature space and the data becomes linearly separable. These examples illustrate the concept and power of the kernel mapping to translate complex separation in low-dimensions to simple separation in high-dimensions.
Figure 3.6-2  XOR example in input space
Figure 3.6-3  Encircled data kernel example
Several kernel functions have been proposed to accomplish this mapping, however not all kernel functions are equally useful. For this study the following two types of kernel functions are considered:

1. Polynomial Kernel

\[ K(x, y) = (x^T y)^d, \text{ where } d > 0 \text{ is a constant} \tag{Equation 3-21} \]

2. Gaussian radial basis function (RBF) kernel

\[ K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right), \text{ where } \sigma > 0 \text{ is a constant} \tag{Equation 3-22} \]

Here \( d \) is the degree of the polynomial kernel and \( \sigma \) the variance of the Gaussian. Both of these kernels satisfy Mercer’s conditions and can be used for mapping into the feature space via an optimal separating hyperplane. In MATLAB, These kernels were coded into a function ‘mykernel’ to perform this analysis and the source code in Figure 3.6-4.
function K = mykernel(ker,x1,x2,arg1,arg2)
% Function that computes the kernel matrix.
%
% Input
% x1,x2 : input vectors
% kernel : kernel function
% Type          Function           Option
% 'linear'  ... linear kernel  k(a,b) = a'*b
% 'poly1'   ... polynomial     k(a,b) = (a'*b)^arg[1]
% 'poly2'   ... polynomial     k(a,b) =
% (a'*b+arg[2])^arg[1]
% 'rbf'     ... RBF (Gaussian) k(a,b) = exp(-0.5*||a-b||^2/arg[1]^2)
%
% Output:
% K [n1 x n1] or K [n1 x n2] Kernel matrix.
%
% Author: Karmon Vongsy
if (nargin < 3) % check correct number of arguments
    help kernel
else
    switch lower(ker)
    case 'linear'
        K = x1'*x2;
    case 'poly1'
        K = (x1'*x2).^arg1;
    case 'poly2'
        K = (x1'*x2 + arg2).^arg1;
    case 'rbf'
        n=size(x1',1);
        K=x1'*x2/arg1^2;
        d=diag(K);
        K=K-ones(n,1)*d'./2;
        K=K-d*ones(1,n)/2;
        K=exp(K);
    end
end

Figure 3.6-4 Kernel MATLAB code

First the kernel is normalized such that any pixel in the data set, k(x,y) = 1. This method is valid because any kernel satisfied by dot products can be normalized and the result remains that of the original kernel. Before proceeding to the dissimilarity measure the
geometry of the feature space is denoted in Figure 3.6-5 with the parameters \( w \) and \( \rho \), defining the hyperplane, \( W \) which is orthogonal to \( \frac{\rho}{\|w\|} \) and also corresponds to the distance from the origin to the hyperplane.

Figure 3.6-5  Kernel feature space geometry

This dissimilarity measure is defined in feature space assuming that two single-class classifiers are trained on both images [18]. The flow chart in Figure 3.6-6 summarizes this algorithm.
To compute the dissimilarity measure in feature space the two sets of data are projected after expanding the space geometry to include two classes. After training two single-class support vectors independently on two sets figure 3.6-7 shows the new feature geometry to include two hyperplanes $W_1$ and $W_2$. 

$H_0: D(x, y) \leq \eta$ (no abrupt change) 

$H_1: D(x, y) > \eta$ (an abrupt change)
Figure 3.6-7 Dissimilarity measure geometry in feature space
For the configuration above, \( c_1 \) is the center of the region corresponding to the subset of interest and \( p_1 \) is a point on the hyperplane. To provide a contrast measure between the two data sets, the calculation of the arc distance between the centers is

\[
D (x, y) = \frac{c_1 \hat{c}_2}{c_1 \hat{p}_1 + c_2 \hat{p}_2}
\]

where,

\( c_1 \hat{c}_2 \) = arc distance between centers \( c_1 \) and \( c_2 \)

and

\( c_1 \hat{p}_1 \) = measure of the spread of samples

Notice that the more the samples are spread, the larger the distance \( c \),\( p \) and the smaller the margin \( \frac{\rho}{\|w\|} \). Expressing the dissimilarity measure in terms of kernels yields

\[
D(x, y) = \frac{\alpha_1^T K_{12} \alpha_2}{\sqrt{\alpha_2^T K_{22} \alpha_2 \rho_1 + \sqrt{\alpha_1^T K_{11} \alpha_1 \rho_2}}}
\]

The dissimilarity measure defined, Equation 3-22 is based on the arc distance in feature space. The derived equations of the kernel dissimilarity measure are included in Appendix A.
4. HSI CAMERA AND DATA

4.1. Hyperspectral Sensor

For this study, the images used for the implementation of the different CD methods were obtained from the sensor Pantera TF 1M30 camera from Cambridge Research & Instrumentation, Inc (CRi) VariSpec™ Liquid Crystal Tunable Filter (LCTF), available at the Wright Patterson Air Force Research Lab (AFRL). The images obtained with the hyperspectral camera were used to test the algorithms under controlled conditions.

The Pantera TF 1M30 camera (Figure 4.1-1) provides high-sensitivity 12 bit images with a 512 x 512 spatial resolution capability at up to 30 frames per second (fps). The Pantera TF 1M30 is a frame transfer charge-coupled device (CCD) camera using a TrueFrame™ progressive scan CCD to simultaneously achieve outstanding resolution and gray scale characteristics. The 12 bit performance provides up to 4096 distinct gray levels. The basic construction in the camera’s image acquisition is an array of values, each representing a grey-scale intensity detected at a particular area in the object plane. The two dimensional square pixel format array is indexed in the sequence corresponding to adjacent points in the object plane. This is often referred to as an image band or frame. Ideally, each pixel is exposed to the object plane simultaneously over the same duration and the values are stored [25].
Figure 4.1-1  Pantera TF 1M60

The LCTF is easily controllable from a computer making an efficient, automated and relatively fast imaging environment. The sensor design optical path is pictured in Figure 4.1-2. Hyperspectral images are acquired by placing the filter in the optical path of the camera and capturing the image through the SMC Pentax 75mm focal lens.
Aperture and exposure duration for each image band were adjusted so that the image data were within the frequency response of the camera (Figure 4.1-3) and the focus was adjusted for each data cube.

![Figure 4.1-3 Pantera TF 1M60 responsivity](image)

The wavelength ranges from 400nm to 720nm in increments of 10nm. Figure 4.1-4 shows measured transmittance of the filter. [26] Multiple images were captured for settings of the tunable filter for the visible region at every 10nm under daylight illumination.
The hyperspectral sensor was interfaced to a controlling computer system provided by Space Computer Corporation (SCC). The PHIRST Light II system [27] was provided for acquisition of visible (Figure 4.1-5) hyperspectral imagery from a liquid crystal tunable filter based camera system.
Figure 4.1-5 Visible Spectrum (www.yorku.ca/eye/spectrum.htm)
The primary functions of the PHIRST Light II system software are to control the camera, frame grabber, and LCTF, to provide live image and statistical information, and to store hyperspectral data to disk. Figure 4.1-6 provides an example of the operator interface and several of the displays available. The system creates HSI cubes by explicitly tuning an optical filter through a sequence of wavelengths, and acquiring one or more image frames at each wavelength. This process captures each frame via a Coreco Imaging PC-CamLink frame grabber, a two-dimensional image over a narrow band of the spectrum is recorded and the third spectral dimension is added over time. The system utilizes a CRI VariSpec™ LCTF to allow the transmission of operator-selected wavelengths of light at chosen intervals along a sweep path. The operator determines the wavelength sequence at initialization, as seen in Figure 4.1-6. This visualization software allows the operator to specify the camera integration time independently for each wavelength in a sweep sequence [26].
Figure 4.1-6  Initialization dialog box
4.2. HSI Data Collection

This study used a series of HSI data to analyze CD techniques. The data cube collection near-to-target sensing consisted of acquiring gray level 512 x 512 images with 33 bands from 400nm to 720nm. For each band, expected dark offset was collected and subtracted from the original. The first data collection shown in Figure 4.2-1(a) and Figure 4.2-1(b) were utilized for cube-to-cube analysis of dismount footprints on grass. The two images visually appears to look alike, however the first data cube (a) was acquired approximately 12 minutes prior to the second cube (b) and then later was collected after two dismounts walked across the viewing scene. The experiment generated image cubes every 5 minutes after the dismounts walked through the scene. The data cubes shown in Figure 4.2-1(c,d) are used to demonstrate CD capabilities from a sequence of images contained in a single HSI cube that contains dismounts strolling in a parking lot. The data cube in Figure 4.2-1 (c) was collected from the roof at Wright-Patterson Air Force Base (WPAFB) throughout the morning of June 28, 2005. Figure 4.2-1 (d) illustrates the HSI cube acquired from the roof top at Wright State University (WSU) on May 29, 2007. This change is created by inter-band analysis of activity monitored in the parking lot. The RGB images in the Figure 4.2-1 were formed using three 10nm HSI bands at 480nm, 510nm and 650nm representing Blue, Green and Red, respectively.
4.3. Spectral Signature Evaluation

An understanding of spectral signatures is essential in the interpretation of a hyperspectral data cube image. Different materials can be discriminated by wavelength-dependent reflection and absorption. Thereby each feature has its own unique spectral reflectance and a graphical representation as a function of wavelength can be created which is defined as its spectral signature. Hyperspectral sensors provide a more
sophisticated approach toward spectral signatures due to the scanning of possible hundreds of closely spaced and very narrow spectral bands to create a continuous spectral response curve. Figures 4.3-1, 4.3-2 and 4.3-3 show the image statistics of the different data image cubes collected as described in the previous section. The plots shown explore the mean, maximum, minimum and standard deviation of the data points. The blue box drawn on the data image illustrates the selected pixels used in investigating the statistics. This method of extracting the spectral signature assesses a potential problem of sensor saturation. The clipping of the plot on the upper limit in Figure 4.3-3 is evident of sensor saturation potentially invalidating some measurements.

Figure 4.3-1 Grass data image statistics
Figure 4.3-2 Parking lot image statistics

Figure 4.3-3 Car data image statistics
5. CHANGE DETECTION SIMULATION RESULTS

5.1. Change Detection using Image Differencing

The simulations of the algorithms were performed in the software environment MATLAB. A normalization technique resulted as an integral step in differencing, ratioing, modified correlation coefficient and kernel dissimilarity measure algorithms. Normalizing is an important process that can improve the results produced during CD. Normalize each frame of the image cube by dividing each element in the image band by the Euclidian norm of the band.

\[
\lambda'(x, y) = \frac{\lambda(x, y)}{\sqrt{\sum_x \sum_y \lambda(x, y)^2}}
\]

Equation 5-1

In these cases, the preprocessing step of normalizing the data is needed to adjust for the different illumination effects within the data in order to create a common basis for CD.

In the image differencing method, for mode 1, the grass data in two collected cubes was used in the differencing CD analysis. The difference of the same frequency bands were change processed to result in a single cube structure. This single cube structure of the same size dimension as both input data cubes is calculated as the absolute difference of frequency band \( \lambda \) from cube 1 and frequency band \( \lambda \).
from cube 2. Because there was no motion during collection all bands provide some
sense of change therefore, neglecting lower frequencies because of the limitations of the
camera response, all bands in the change cube were summed. Figure 5.1-1 illustrates the
summation of the changed cube bands displaying tracks of two dismounts intruding after
image cube 1 was collected.
Figure 5.1-1  Differencing summation change
The difference map is binarized by thresholding to obtain a binary image. The choice of threshold value is critical, because false detection could result or a true change may not get detected. A threshold was applied to the raw summation difference image at multiple levels in an intermediate range of intensities guided by the minimum and maximum generated values.

### 5.2. Change Detection using Image Ratioing

This section presents the results obtained with hyperspectral images in which change was evaluated using the image ratioing procedure. Similar to the image differencing technique images are compared pixel-by-pixel by a ratio algorithm. The result for the cube-to-cube mode of the image ratioing procedure is shown in Figure 5.2-2. This figure depicts the ratio image which again is compiled into a 3D image array and the summation of the change information. The main assumption of this technique is that without significant spectral change the ratio between two images will result in unity. Thus change can be observed with a higher or lower ratio value.
Figure 5.2-1 Ratioing summation change
After analyzing the change information cube a binary change cube was constructed. The threshold value decision differs slightly from the differencing method because it is derived from a multilevel threshold. The threshold values were determined in either direction from the unity value found by experimentation.

5.3. Change Detection using PCA

The visual results obtained for PCA are presented below. As discussed previously this method results in two cubes after processing. Referring to Figure 5.3-1, the second cube corresponds to the second principal component which displays the change results. The pixel values for each image cube were vectorized and then combined in a matrix before obtaining the principal components of this new matrix. Figure 5.3-1 presents the restructured change results cube obtained from the cube-to-cube grass data analysis. As can be seen the algorithm was capable of detecting changes from the dismount tracks in the summed result CD cube.
Figure 5.3-1  PCA summation change
Figure 5.3-2, which used visual interpretation to decide on a threshold that best highlights the change. The white (1) represents the change or in this study the dual set of tracks entering and leaving the scene.
Figure 5.3-2  PCA after thresholding
5.4. Linear Chronochrome Change Detection Results

The linear chronochrome CD algorithm first subtracts the mean from both data cubes for implementing cube-to-cube mode. The Chronochrome algorithm implements a 3X3 pixel-size sliding window before calculating the multivariate statistics of the two data sets. After applying a least mean-squared error predictor and summing all bands, Figure 5.4-1 exposes the dismount track change. An anomaly detector is then applied to the error image by using Mahalanobis distance. After applying this technique to identify the anomalous changes, the result is showcased in Figure 5.4-2. A threshold is applied to represent the binary image in Figure 5.4-3.

![Summation of change cube](image)

Figure 5.4-1 LCC error prediction
Figure 5.4-2  LCC after applying anomaly detection
5.5. Maximum-Likelihood Change Detection Results

As was mentioned in the previous chapter, this algorithm is based on the Maximum-Likelihood CD algorithm, modified to incorporate hyperspectral data. The function derived for this procedure utilizes a sliding window processing specified by the user to perform the needed calculations. The window size chosen to produce the results shown is a 3X3 pixel size. Figure 5-8 shows footprints-on-grass results produced by applying the correlation coefficient algorithm in cube-to-cube mode to reveal the temporal change occurring in a particular scene between the acquisition times of two data cubes. This resultant correlation coefficient is a summation of all wavelength results attained from the two input data cubes. The change image in Figure 5.5-1 is a two
dimensional image showing where the reflectivity of the grass has remained the same or been change between image cube collections.
Figure 5.5-1  Correlation coefficient summation change
To help eliminate false alarms, a threshold process is subjectively decided based on the change result image revealed. This binary thresholded image can be seen in Figure 5.5-2.
Figure 5.5-2 Correlation Coefficient with Thresholding
5.6. Change Detection using Kernel Approach

For the initial kernel dissimilarity experiment, a second-order mapping polynomial was used, i.e.,

\[
K(x, y) = (x^T y)^d
\]

Equation 5-2

along with parameters \( \rho_1 = \rho_2 = 0.5 \) in Equation 3-17. This algorithm applied a sliding 3X3 pixel window and excluded the lower frequency bands of 400-460 nm to lower the noise response. Results of the processing for this experiment are shown in Figure 5.6-1. The longest wavelength band corresponding to 720 nm was considered to hold the most distinguishable change map and therefore contained the change kernel dissimilarity measure for the cube-to-cube mode. The threshold binary change map for the polynomial kernel is represented in Figure 5.6-2.
Figure 5.6-1  Polynomial kernel dissimilarity measure change map

Figure 5.6-2  Threshold polynomial kernel dissimilarity measure change map
In contrast, the kernel dissimilarity algorithm was also configured using the Gaussian radial basis function kernel

\[
K(x, y) = \exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)
\]

RBF: \hspace{1cm} \text{Equation 5-3}

where \( \sigma = 0.5 \) was subjectively chosen as a parameter. The result of the RBF kernel parameter effect is shown in Figure 5.6-3. Figure 5.6-4 displays the threshold results of the different kernel parameter examined.

Figure 5.6-3 RBF kernel dissimilarity measure change map
The above results were for cube-to-cube mode, visualizing the characteristics of the two dismount foot tracks across the grass scene.

The minimum and maximum values of these results as well as the threshold values employed on the data are summarized in Table 5-1.

<table>
<thead>
<tr>
<th>Method</th>
<th>Min</th>
<th>Max</th>
<th>Threshold (T)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differencing</td>
<td>50</td>
<td>6885</td>
<td>T &gt; 1800</td>
</tr>
<tr>
<td>Ratioing</td>
<td>11.1245</td>
<td>65.9239</td>
<td>17 &gt; T &gt; 35</td>
</tr>
<tr>
<td>PCA</td>
<td>-4658.8</td>
<td>4658.8</td>
<td>-1200 &gt; T &gt; 1200</td>
</tr>
<tr>
<td>LCC</td>
<td>5.5289e+013</td>
<td>1.2714e+020</td>
<td>T &gt; 2e+18</td>
</tr>
<tr>
<td>Correlation coeff.</td>
<td>13.0753</td>
<td>16.4518</td>
<td>T &gt; 15.85</td>
</tr>
<tr>
<td>Polynomial kernel</td>
<td>0.9694</td>
<td>1.0354</td>
<td>T &gt; 1.009</td>
</tr>
<tr>
<td>RBF kernel</td>
<td>0.9999999999624</td>
<td>1.000000000000001</td>
<td>T &gt; 0.999999999999624</td>
</tr>
</tbody>
</table>

Table 5-1 Results of cube-to-cube simulation
5.7. Inter-band Results using Correlation Coefficient

Results obtained by applying correlation coefficient algorithm to exploit changes occurring in the dwell time between successive bands in a given data cube are presented in this section. Figure 5.7-1 depicts the correlation coefficient CD results revealing the individual dismounting the vehicle though saturation exists. Figure 5.7-2 exposes the change of three individuals walking through the parking lot. In this experiment bands 1-5 were eliminated because of the noise indicative of the blue frequency region camera response.
Figure 5.7-1  Correlation coefficient results for WPAFB car data collection
5.8. Time Lapse Results on Grass Data

This study was designed to access the CD for different intervals on the grass-track collection. The grass data collection time before tracks was time-stamped at 10:51:19. The scene was then marked by two dismount foot-tracks entering and leaving the scene. The time recorded for collection of this experiment is summarized in Table 5-2.
Table 5-2  Summary of grass data

<table>
<thead>
<tr>
<th>Data</th>
<th>Time-stamp (hr:min:sec)</th>
<th>Time Difference (hr:min:sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior to dismount tracks</td>
<td>10:51:19</td>
<td></td>
</tr>
<tr>
<td>After dismount tracks</td>
<td>10:57:51</td>
<td>00:06:32</td>
</tr>
<tr>
<td>After dismount tracks</td>
<td>11:02:20</td>
<td>00:11:01</td>
</tr>
<tr>
<td>After dismount tracks</td>
<td>11:08:06</td>
<td>00:16:47</td>
</tr>
</tbody>
</table>

The results below indicate the results of the modified correlation coefficient for each of the three differences of time.

Figure 5.8-1  Change after 6:32 minutes
Figure 5.8-2  Change after 11:01 minutes

Figure 5.8-3  Change after 16:47 minutes
Visually it can be seen that after several minutes the change deteriorates and the tracks from the dismounts are less obvious.

To quantify assess the performance of the CD algorithms, use was made of a receiver operating characteristic curve (ROC) obtained by plotting the correct detection probability, \( P_D \) against the false alarm probability \( P_{FA} \) based on a devised truthing map for the dual set of grass-foot-prints. The area under the ROC curve determines the level of competency in the correct classification. A value of 0.5 suggests a mere hazarding of guesses whilst a value of 1 indicates correct classification every time.

To detect a change, an empirical threshold was applied to the changed map, this threshold value is critical because this affects how many changed pixels are missed and how many unchanged pixels are falsely detected. To avoid this issue of comparing thresholds a ROC curve is employed. This methodology is evaluated for each threshold value in a range whilst calculating the percentage \( P_D \) of changed pixels were positively detected and \( P_{FA} \) of unchanged pixels classified as changed. The ROC curve is a plot of \( (P_{FA}, P_D) \) values as the threshold changes. This will show the range of performances possible, without having to directly select comparable threshold values. This procedure allows a satisfactory tool for comparison.

A ground truth to calculate \( P_{FA} \) and \( P_D \) is needed to determine consequence rates between the binary ground truth map and each detection result. The truthing map in Figure 5.8-4 was generated by manually selected the change region where,

\[
T(x, y) = \begin{cases} 
1, & \text{true pixel location} \\
0, & \text{elsewhere}
\end{cases}
\]  

Assuming \( D \) are properly matched after desired threshold is implemented then the probability of detection is
\[ P_D = \frac{D}{T} \]  \hspace{1cm} \text{Equation 5-5}

and the probability of false alarm is

\[ P_{FA} = \frac{M - D}{N - T} \]  \hspace{1cm} \text{Equation 5-6}

where \( N \) = total number of pixels in image and \( M \) = declared change for a particular threshold from the algorithm. \( P_{FA} \) represents the pixels that weren’t actual change but got falsely detected as change. The \((P_{FA}, P_D)\) pair is one point on the ROC curve. This process is repeated for all possible threshold values which develop the curve. During generation, the values \( M \) and \( D \) will vary with various thresholds.
Figure 5.8-4 Dismount tracks truthing map

The ROC curves for all implemented CD algorithms are shown in Figure 5.8-5. This plot details the ROC observation using the ground truth of Figure 5.8-4. In these curves, the smaller thresholds are at the bottom-left portion of the curves increasing to a maximum at the upper right corner.
Figure 5.8-5 ROC curve plots

The ROC plot of Figure 5.8-5 shows the modified correlation coefficient outperforms the other simulated CD. The ROC plot for the study on time differences involving the dismount tracks is documented in Figure 5.8-6.
As the above plot displays, detection diminishes with each step in time lapse. A dramatic effect is seen after 16:47 minutes.
6. CONCLUSIONS AND FUTURE WORK

6.1. Concluding Summary

This thesis set out to compare and explore Change Detection for hyperspectral imagery based on temporal and spectral changes between successive observations. The work in this study surveyed CD by considering close-in observed hyperspectral surveillance relative to motion in a scene.

CD depends on temporal and spectral effects and the variability and analysis of the spectral signature can aid in the evaluation relative to change. Based on the visual and quantitative results presented above, some initial conclusions can be drawn about the effectiveness of the various CD algorithms as applied to the particular data sets considered in this thesis. A visual examination of each of the resulting output plots indicates that the modified correlation coefficient method appears to be most effective in revealing the changes in the scene under examination relative to the other algorithms that were considered. The ROC plot of Figure 5.8-5 supports this conclusion. Based on the background research conducted in this effort, it was hypothesized that the kernel algorithm would yield superior results than linear algorithms. However, in our limited subjective results with minor parameter tuning, as implemented in this research did not support this theory.

Comparative results, in terms of computational complexity are shown in Table 6-1, where computational time estimates for each algorithm to run change detection results
are reported. When selecting an appropriate algorithm for a given application, computational complexity must be considered.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Grass Data (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Differencing</td>
<td>23.032</td>
</tr>
<tr>
<td>Ratioing</td>
<td>55.078</td>
</tr>
<tr>
<td>PCA</td>
<td>77.953</td>
</tr>
<tr>
<td>LCC</td>
<td>513.031</td>
</tr>
<tr>
<td>Correlation Coefficient</td>
<td>291.156</td>
</tr>
<tr>
<td>Kernel (polynomial)</td>
<td>9758.531</td>
</tr>
<tr>
<td>Kernel (RBF)</td>
<td>11700.922</td>
</tr>
</tbody>
</table>

Table 6-1 Computation time of change detection algorithms

The computational burden of the kernel algorithm superseded and dominated the results. This method is quite expensive from the computational load perspective assumed by the mapping of data from input space into the higher dimensional feature space requirement by the Kernel based algorithm. Thereby this approach may not be well suited for applications with fast execution requirement.
6.2. Future Work

Future research on new data under controlled experiments with various types of changes testing the algorithms presented should further support these findings. An interesting comparison of algorithm result accuracy with an alternate RGB sensor could provide the benefits of HSI data. An investigation and research on threshold techniques for the selection of optimal threshold values would eliminate subjective visual results. For advanced applications and future algorithm development, where complex environmental conditions such as atmospheric propagation or camera variances are to be included in the model, appropriate preprocessing algorithms should be considered. The change map is prone to changes in light and shadow differences; eliminating such false positives is a major area in future work. Further work is also needed on kernelization of Linear CD algorithms presented in this thesis. It is conceivable that different choice of kernels or different set of parameter choices may produce more meaningful CD results. The data implemented in this thesis was constrained to no sensor movement, meaning image registration was not a concern. However, for future experiments, covariance equalization is a method to aid ill-registered images described by Schaum and Stoker [28].
APPENDIX A

A.1 Derivation of the maximum likelihood estimate

A derivation of the maximum likelihood estimate of the degree of change in two
HSI data collections is presented. The HSI cubes or frames are vectorized into long
vectors and the measured reflectivity at pixel position \( n \) in two vector images are denoted
by \( x_n \) and \( y_n \). These measured values are represented by the models,

\[
\begin{align*}
    x_n &= s_n + n_1 \\
    y_n &= \alpha s_n + (\sqrt{1-\alpha^2})z_n + n_2
\end{align*}
\]

where, \( s_n \) denotes the true reflectivity and \( n_1 \) and \( n_2 \) are additive noise terms.

\[
\eta = \begin{bmatrix} x_n \\ y_n \end{bmatrix} = \begin{bmatrix} s_n \\ \alpha \end{bmatrix} + z_n \begin{bmatrix} 0 \\ \sqrt{1-\alpha^2} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \end{bmatrix}
\]

Under the assumption that the random variables are zero-mean, Gaussian and mutually
independent the conditional probability density function (PDF) follows as :

\[
f(\eta | \alpha) = \frac{1}{\pi^2 |Q|} \exp(\eta^T Q^{-1} \eta)
\]
where,

\[
Q = E\{\eta \eta^T\} = \begin{bmatrix}
\sigma_s^2 + \sigma_n^2 & \alpha \sigma_s^2 \\
\alpha \sigma_s^2 & \sigma_s^2 + \sigma_n^2
\end{bmatrix}
\]

The likelihood function can be computed by taking the logarithm and deleting any terms that are not a function of \(\alpha\), the mathematics is presented.

\[
f = \ln \frac{1}{c |Q|} - \ln \left[ -X^T Q^{-1} X \right]
\]

\[
f = \ln \frac{1}{c \left[ \sigma_s^2 + \sigma_n^2 \frac{\alpha \sigma_s^2}{\alpha \sigma_s^2 + \sigma_s^2 + \sigma_n^2} \right]^N} - \sum_{k=1}^N \left[ g_k \ h_k \left[ \sigma_s^2 + \frac{\alpha \sigma_s^2}{\sigma_s^2 + \sigma_n^2} \right] \right] \left\{ x_k \right\} \left\{ y_k \right\}
\]

\[
f = \ln \frac{1}{c \left[ (\sigma_s^2 - \alpha^2 \sigma_s^4 + 2\sigma_n^2 \sigma_s^2 + \sigma_n^4)^N \right]} - \sum_{k=1}^N \left\{ \frac{-2x_k y_k \alpha + y_k^2 + x_k^2}{(1 - \alpha^2) \sigma_s^2} \right\}
\]

\[
f = -cN \ln \left[ \sigma_s^4 (1 - \alpha^2) + 2\sigma_n^2 \sigma_s^2 + \sigma_n^4 \right] - \sum_{k=1}^N \left\{ \frac{(x_k^2 - 2x_k y_k \alpha + y_k^2)}{(1 - \alpha^2) \sigma_s^2} \right\}
\]

The next step in finding \(\alpha\) that maximizes the above function is to differentiate the expression with respect to \(\alpha\) and set to zero; solving for \(\alpha_{ML}\) yields the maximum likelihood coefficient used in CD.

\[
0 = 2N \alpha \sigma_s^4 + \frac{2g_k h_k}{(\alpha^2 - 1) \sigma_s^2} + \frac{2(g_k^2 - 2g_k h_k \alpha + h_k^2) \alpha}{(\alpha^2 - 1)^2 \sigma_s^2}
\]

Finally, the Maximum likelihood estimator of \(\alpha\) is given by,

\[
\alpha_{ML} = \frac{2 \cdot \sum_{i} x_i y_i}{\sum_{i} x_i^2 + \sum_{i} y_i^2}
\]
A.2 Derivation of the kernel dissimilarity measure

A derivation of the kernel dissimilarity measure is provided for the algorithm development involving kernelization. As explained by [19] the dissimilarity measure for comparing two sets of data via arc distances in feature space is

\[ D(x, y) = \frac{c_1 \tilde{c}_2}{c_1 \tilde{p}_1 + c_2 \tilde{p}_2} \]

where,

\[ c_1 \tilde{c}_2 = \text{arc distance between centers } c_1 \text{ and } c_2 \]

and

\[ c_1 \tilde{p}_1 = \text{measure of the spread of samples} \]

In the context of this work, using the dissimilarity measure, the derived equations of the kernel dissimilarity measure for change in a pair of HSI collections is provided.
\[ D(x, y) = \frac{c_1 \hat{c}_2}{c_1 \hat{p}_1 + c_2 \hat{p}_2} \]

\[ c_1 \hat{c}_2 = \frac{\alpha_1^T K_{12} \alpha_2}{\sqrt{\alpha_1^T K_{11} \alpha_1} \sqrt{\alpha_2^T K_{22} \alpha_2}} \]

\[ c_1 \hat{p}_1 = \frac{\rho_1}{\sqrt{\alpha_1^T K_{11} \alpha_1}} \quad c_2 \hat{p}_2 = \frac{\rho_2}{\sqrt{\alpha_2^T K_{22} \alpha_2}} \]

\[ D(x, y) = \frac{\rho_1}{\sqrt{\alpha_1^T K_{11} \alpha_1}} + \frac{\rho_2}{\sqrt{\alpha_2^T K_{22} \alpha_2}} \]

Let \( a = \sqrt{\alpha_1^T K_{11} \alpha_1} \) and \( b = \sqrt{\alpha_2^T K_{22} \alpha_2} \)

\[ D(x, y) = \frac{ab}{a + b} \]

\[ D(x, y) = \frac{ab}{y\rho_1 + x\rho_2} = \frac{\alpha_1^T K_{12} \alpha_2}{ab} \cdot \frac{ab}{y\rho_1 + x\rho_2} \]

\[ D(x, y) = \frac{\alpha_1^T K_{12} \alpha_2}{b\rho_1 + a\rho_2} \]

substituting \( a \) and \( b \) back into equation

\[ D(x, y) = \frac{\alpha_1^T K_{12} \alpha_2}{\sqrt{\alpha_2^T K_{22} \alpha_2} \rho_1 + \sqrt{\alpha_1^T K_{11} \alpha_1} \rho_2} \]
A.3 Mercer’s Theorem

As stated in this thesis a legitimate kernel must satisfy Mercer’s theorem. Therefore, Mercer’s theorem says when a function $K$ is a kernel. The theorem is summarized for review in this section.

**Theorem:** Any function $K(x, y)$ that is symmetric non-negative definite can be expressed as a dot product in a high-dimensional space. If

$$\sum_{i,j} K(x_i, x_j)c_i c_j \geq 0$$

for every finite subset $\{x_1, ..., x_n\}$ of $X$ and every subset $\{c_1, ..., c_n\}$ of real numbers, then there exists a function $\varphi(x)$ such that $K(x, y) = \varphi(x) \cdot \varphi(y)$. A symmetric positive definite matrix has positive eigenvalues [24].
APPENDIX B

B.1 MATLAB code for implementation of Principal Component Analysis
This code was used in the implementation of PCA for CD.

function [pc1 pc2] = image_pca(im1,im2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% This function takes in a hyperspectral data cube as its input and
% outputs a data cube whose frames show where change is detected.
% % REQUIRED INPUTS:  im1 = hyperspectral data set 1
% % im2 = hyperspectral data set 2
% % OUTPUTS:          pc1 = first principal component
% % pc2 = second principal component (change info)
% % AUTHOR:  Karmon M. Vongsy
%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% getting the size of the data cube
[m,n,o] = size(im1);
% mean of data
for ind2 = 1:o
im1(:,:,ind2) = im1(:,:,ind2) - ones(512)*mean(mean(im1(:,:,ind2)));
end
Im_1 = reshape(im1,m*n,o);
clear im1
[m,n,o] = size(im2);
%%% mean of data
for ind2 = 1:o
im2(:,:,ind2) = im2(:,:,ind2) - ones(512)*mean(mean(im2(:,:,ind2)));
end
Im_2 = reshape(im2,m*n,o);
clear im2

[p,q] = size(Im_1);
V1 = reshape(Im_1,p*q,1);
V2 = reshape(Im_2,p*q,1);
Mat = [V1 V2];

[pc,EigVal,Percentages] = pcacov(cov(Mat));
%%% transforming data
\[ T = \text{Mat}\cdot\text{pc}; \]

\[ P1 = T(:,1); \]
\[ P2 = T(:,2); \]

%% reshape to original size
\[ pc1 = \text{reshape}(P1, m, n, o); \]
\[ pc2 = \text{reshape}(P2, m, n, o); \]

%Change Information (Second Component)
B.2 MATLAB code for the Linear Chronochrome Change detection

The linear chronochrome functions perform the computation of the error matrix and a Mahalanobis anomaly detector.

```matlab
function [c_map] = LCC_change_det(data_cube, type, b)

% This function takes in a hyperspectral data cube as its input and outputs a data cube whose frames show where change is detected.
% REQUIRED INPUTS: data_cube = hyperspectral data set
% type = 1, cube to cube analysis
% 2, single cube analysis
% b = block size, should be odd number
% OUTPUTS: c_map = changed output
% AUTHOR: Karmon M. Vongsy

% getting the size of the data cube
[m, n, o] = size(data_cube);
b = b - 1;

% mean subtraction of data
for ind2 = 1:o
    data_cube(:, :, ind2) = data_cube(:, :, ind2) - ones(512) * mean(mean(data_cube(:, :, ind2))); % This line can be optimized for speed.
end

if ( (type == 1) & ~mod(o, 2) )
    o = o/2;
    shifted = o;
    ind_add = 0;
elseif ( (type == 1) & mod(o, 2) )
    error('type is wrong or the data cube is wrong')
else
    shifted = -1;
    ind_add = 1;
end

% initializing c_map
% c_map: change matrix

c_map = zeros([(m-b) (n-b) (o - ind_add)]);

% obtaining the correlations between each adjacent frame
for ind1 = (ind_add + 1):o
    for ind2 = 1:(m-b)
        for ind3 = 1:(n-b)
```
%%% each sample bxb block is taken and vectorized
x = data_cube(ind2:(ind2+b),ind3:(ind3+b),ind1);
x = x(:);
y = data_cube(ind2:(ind2+b),ind3:(ind3+b),ind1 + shifted);
y = y(:);
%%% compute covariance matrices
C = y'*x;
x = x'*x;
Y = y'*y;
y_hat = C*inv(X)*x;
%%% compute error signal
E_CC = y - y_hat;
c_map(ind2,ind3,(ind1-ind_add))= E_CC;
end
end
end

function out = anomaly_dector(in)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Computes the Mahalanobis distance between each pixel to the nearest
% mean cluster which is used as the anomaly score
%%%%
% REQUIRED INPUTS
% in:      image input
%
% OUTPUTS
% out:       2D M-distance map
%
% Author: Karmon M. Vongsy
%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Size and reshape
[ sy, sx, sz ]=size( in );
in = double( reshape( in, sy*sx, sz ) );

%%% Compute M-distances
in = in - repmat( mean(in), sx*sy, 1 );

if sz > 1
   out = sum ( ( ( in * cov(in) ) .* in )' );
else
   out = ( ( in * cov(in) ) .* in )';
end

%%% Reshape into 2D image
out = reshape( out, sy, sx );
B.3 MATLAB code for implementation of Modified Correlation Coefficient

MATLAB was used to generate a function capable of performing the modified correlation coefficient algorithm. The code was used to implement the CD process described in section 3.5.

```matlab
function [c_map, cor_map] = hyper_change_det(data_cube, type, b, threshold)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% This function takes in a hyperspectral data cube as its input and
% outputs a data cube whose frames show where change is detected.
% % REQUIRED INPUTS:
% data_cube = hyperspectral data set
% type = 1, cube to cube analysis
% 2, single cube analysis
% threshold = desired threshold to compute binary
% image
% b = block size, should be odd number
% OUTPUTS:
% cor_map = actual values of changed output
% c_map = thresholded values of changed output
% %
% % AUTHOR: Karmon M. Vongsy
% % initially created by Andrew Kondrath
%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[m,n,o] = size(data_cube);
b = b-1;

for ind = 1:o
    data_cube(:,:,ind) = data_cube(:,:,ind)/sqrt(sum(sum(abs(data_cube(:,:,ind)).^2)));
end

if ( (type == 1) & ~mod(o,2) )
    o = o/2;
    shifted = 0;
    ind_add = 0;
elseif ( (type == 1) & mod(o,2) )
    error('type is wrong or the data cube is wrong')
else
    shifted = -1;
    ind_add = 1;
end

%% initializing cor_map and c_map
%% cor_map: holds the correlation values of each 3x3 block
%% c_map: truth matrix of which correlations are less than a set
%% threshold
```
cor_map = zeros([(m-b) (n-b) (o - ind_add)]);
c_map = zeros([(m-b) (n-b) (o - ind_add)]);

%%% obtaining the correlations between each adjacent frame

for ind1= (ind_add + 1):o
    for ind2= 1:(m-b)
        for ind3= 1:(n-b)
            %% each sample 3x3 block is taken and vectorized (for
            %% vector multiplication)
g= data_cube(ind2:(ind2+b),ind3:(ind3+b),ind1);
g= g(:);
h= data_cube(ind2:(ind2+b),ind3:(ind3+b),ind1 + shifted);
h= h(:);
            %% the equation calls for multiplication by 2 for
            %% correlations
            %% between 0 and 1, but for speed, the multiplication is
            %% left
            %% out and the correlation threshold is divided by 2
            c_map(ind2,ind3,(ind1-ind_add))= g'*h / (g'*g + h'*h);
cor_map(ind2,ind3,(ind1-ind_add))= g'*h / (g'*g + h'*h);
        end
    end
end

%%% finding where the correlations are below the desired threshold and
%%% setting those positions equal to 1
pos_cors= find(cor_map>=threshold);
[m n o] = size(c_map);
c_map=ones(m,n,o);
c_map(pos_cors)=0;
B.4 MATLAB code for implementation of the kernel dissimilarity measure

MATLAB was used to generate a function capable of performing the kernel dissimilarity measure algorithm. The code was used to implement the CD process described in section 3.6.

```matlab
function [c_map]= hyper_change_kernel(data_cube,rho1,rho2)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%% This function takes in a hyperspectral data cube as its input and
% outputs a data cube whose frames show where change is detected.
%%%%% REQUIRED INPUTS:  data_cube = hyperspectral data set
%                      rho1 and rho2 from dissimilarity measure
%                      D =    alpha_1'*Kernel_12*alpha_2
%  --------------------------------------------
%           sqrt(alpha_2'*Kernel_22*alpha_2)*rho1 +
%          sqrt(alpha_1'*Kernel_11*alpha_1)rho2
% OUTPUTS:          c_map = changed output
%%%%% AUTHOR:  Karmon M. Vongsy
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% getting the size of the data cube
[m,n,o]= size(data_cube);

% normalizing each frame in the data cube
for ind= 1:o
  data_cube(:,:,ind)= data_cube(:,:,ind)/...  
    sqrt(sum(sum(abs(data_cube(:,:,ind)).^2)));
end

% initializing c_map
% c_map: holds the change values of each 3x3 block
c_map= zeros([(m-2) (n-2) (o-1)]);

% obtaining the correlations between each adjacent frame
for ind1= 2:o
  for ind2= 1:(m-2)
    for ind3= 1:(n-2)
      g = data_cube(ind2:(ind2+2),ind3:(ind3+2),ind1);
      g = g(:)';
      h = data_cube(ind2:(ind2+2),ind3:(ind3+2),ind1-1);
      h = h(:)';
      c_map(ind2:(ind2+2),ind3:(ind3+2),ind1)=...  
        g'*h;
    end
  end
end
```

98
[dim,num_data] = size(g);
%% apply polynomial kernel
    K1 = mykernel('poly1',g,g,2);
%% apply rbf kernel
K1 = myKernel('rbf',g,g,0.5);
for ind = 1:(num_data)
    K1(:,ind)= K1(:,ind)/norm(K1(:,ind));
end

%%% eigen decomposition of the kernel matrix
[U,D] = eig(K1);
Lambda=real(diag(D));

%%% normalization of eigenvectors to be orthonormal
for k = 1:num_data,
    if Lambda(k) ~= 0,
        U(:,k)=U(:,k)/sqrt(Lambda(k));
    end
end

%%% Sort the eigenvalues and the eigenvectors in descending order
[Lambda,ordered]=sort(-Lambda);
Lambda=-Lambda;
U=U(:,ordered);

%%% use first new_dim principal components as weights
A1=abs(U(:,1));

%%% apply polynomial kernel
    K2 = mykernel('poly1',h,h,2);
%%% apply rbf kernel
K2 = myKernel('rbf',h,h,0.5);
for ind = 1:(num_data)
    K2(:,ind)= K2(:,ind)/norm(K2(:,ind));
end

%%% eigen decomposition of the kernel matrix
[U,D] = eig(K2);
Lambda=real(diag(D));

%%% normalization of eigenvectors to be orthonormal
for k = 1:num_data,
    if Lambda(k) ~= 0,
        U(:,k)=U(:,k)/sqrt(Lambda(k));
    end
end

%%% Sort the eigenvalues and the eigenvectors in descending order
[Lambda,ordered]=sort(-Lambda);
Lambda=-Lambda;
U=U(:,ordered);
%%% use first new_dim principal components as weights
A2=abs(U(:,1));

%%% apply polynomial kernel
% K12 = mykernel('poly1',g,h,2
%%% apply rbf kernel
K12 = mykernel('rbf',g,h,0.5);

for ind = 1:(num_data)
    K12(:,ind)= K12(:,ind)/norm(K12(:,ind));
end
c_map(ind2,ind3,(ind1-1))=(A1'*K12*A2)/(rho1*(sqrt(A1'*K1*A1)) + 
    rho2*(sqrt(A2'*K2*A2)));
end
end
end
B.5 MATLAB code ROC curve generation for multiple time lapse grass experiment

This code was used for making the ROC curves in section 5.8. The modified correlation coefficient algorithm was executed and the cube-to-cube changes over all frequencies were summed before saving as a .mat file. A truthing image was formed by selecting the truth-regions using the hyperspectral GUI, and saved as a load_truth.mat file for loading in the code later. The following code then plots each ROC curve and a hold is placed on the figure before to generate ROC curves for other methods. This is done by loading the corresponding change map file for an algorithm (or for the same algorithm but between different time stamps). This code can be modified for different change algorithm files by making an adjustment in the threshold values.

```matlab
%%% Roc analysis for modified correlation coefficient algorithm
%%% for different grass time collection
%%% Karmon M. Vongsy

clear all

%%% load change map file
% load 10_51_19_vs_10_57_51.mat
% load 10_51_19_vs_11_02_20.mat
load 10_51_19_vs_11_08_06.mat

j=1;
i=0;

%%% find minimum and maximum of summation results from modified
correlation coefficient change algorithm
b=min(min(change_map));
e=max(max(change_map));

%%% loop thru threshold values
for i=b:0.00001:e
    threshold=i;
    image_truth=find(change_map>=threshold);

    %% form background matrix of zeros
    decide= zeros(510);
```
%% fill in ones where correlations are above threshold
decide(image_truth)= 1;

%% load truth matrix file chosen by hand
load load_truth.mat
difference_mat=(2.*truth)-decide;

%% properly detected change
D_matched=find(difference_mat==1);
D_size=length(D_matched);

%% actual changes from truth map
T_mat_ones=find(truth==1);
T_mat_size=length(T_mat_ones);

%% calculate probability of detection
P_D(j)=D_size/T_mat_size;
M=find(decide==1);
M_size=length(M);
N=510*510;

%% calculate probability of false alarms
P_FA(j)=(M_size-D_size)/(N-T_mat_size);
j=j+1;
end

%% view plot with varied thresholds
figure;plot(P_FA,P_D);title('ROC Curve');axis([0 1 0 1]);
REFERENCES


22. [http://www.flipturn.org/peppy/Hyperspectral.html](http://www.flipturn.org/peppy/Hyperspectral.html)


