ARMY RESEARCH LABORATORY

Hyperspectral Imaging and Obstacle Detection for Robotics Navigation

by Heesung Kwon, Dalton Rosario, Neelam Gupta, Matthew Thielke,
Dale Smith, Partick Rauss, Patti Gillespie, Nasser M. Nasrabadi

ARL-TR-3639 **September 2005**

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

Citation of manufacturers' or trade names does not constitute an official endorsement or approval of the use thereof.

DESTRUCTION NOTICE—Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory
Adelphi, MD 20783-1145

ARL-TR-3639 **September 2005**

Hyperspectral Imaging and Obstacle Detection for Robotics Navigation

Heesung Kwon, Dalton Rosario, Neelam Gupta, Matthew Thielke,
Dale Smith, Partick Rauss, Patti Gillespie, Nasser M. Nasrabadi

Sensors and Electron Devices Directorate, ARL

REPORT DOCUMENTATION PAGE	Form Approved OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2005	Final	08/2004–03/2005

4. TITLE AND SUBTITLE

Hyperspectral Imaging and Obstacle Detection for Robotics Navigation

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Heesung Kwon, Dalton Rosario, Neelam Gupta, Matthew Thielke, Dale Smith, Partick Rauss, Patti Gillespie, Nasser M. Nasrabadi

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

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

U.S. Army Research Laboratory
Sensors & Electron Devices Directorate
(ATTN: AMSRD-ARL-SE-SE) hkwon@arl.army.mil
Adelphi, MD 20783-1145

8. PERFORMING ORGANIZATION REPORT NUMBER

ARL-TR-3639

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

ARL
2800 Powder Mill Road
Adelphi, MD 20783-1145

10. SPONSOR/MONITOR'S ACRONYM(S)

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

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Recently, object detection based on hyperspectral sensors in support of autonomous robotics navigation has been of great interest. Hyperpspectral sensors have been widely used for automatic target detection in military applications, mainly because a wealth of spectral information can be obtained through a large number of narrow contiguous spectral channels (often over a hundred). The main purpose of this report is to present detection techniques based on hyperspectral sensing that can effectively identify potentially harmful objects to UGV navigation. The hyperspectral detection techniques used are built on the basic premise that the spectral signatures of objects of interest are in general different than background materials, and the objects of interest can be identified from their surrounding background materials based on spectral analysis of the hyperspectral data. In this report, we first present detailed information on two hyperspectral sensors—a dual band hyperspectral imager and an acousto-optic tunable filter imager—that provide hyperspectral data in the infrared and visible bands, respectively. Several anomaly detection and classification techniques newly developed by ARL are then introduced and applied to the hyperspectral data to identify potential obstacles to robotics navigation. Detection performance for each technique is included in this report.

15. SUBJECT TERMS Robotics navigation, obstacle detection, UGV, target detection, hyperspectral.

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Heesung Kwon
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	83	19b. TELEPHONE NUMBER (*Include area code*) (301) 394-1678

Standard Form 298 (Rev. 8/98)

Contents

Figures

1. Introduction

High performance obstacle detection techniques, such as anomaly or target detection, based on imaging sensor data can enhance UGVs' capability to safely maneuver unknown terrain with increased speed and to conduct successful scouting missions without being interrupted or damaged. However, conventional detection techniques based on broadband imaging sensors, such as the FLIR (Forward Looking InfraRed) sensor, utilize only spatial signatures of objects significantly limiting detection performance. Recently, hyperspectral sensors which can operate at various spectral bands (the visible, near IR, or IR band) have been increasingly used for ATR (Automatic Target Recognition) applications mainly because of a wealth of spectral as well as spatial information available in hyperspectral sensor data. Reflectance (or emission) spectra of materials generated by the hyperspectral sensors provide critical information for material classification, as each material type is, in general, uniquely characterized by its corresponding spectrum. The hyperspectral imaging sensor mounted on the UGV is therefore expected to greatly improve the performance of obstacle detection by providing high-resolution spectral profiles of materials within the neighboring areas of the UGV.

In support of robotics autonomous navigation applications, three types of hyperspectral imaging sensors have been used for obstacle detection—a Dualband Hyperspectral Imager (DBHSI), a visible to near-IR spectral imager (SOC-700), and an Acousto-Optical Tunable Filter (AOTF) imager, called SECOTS. DBHSI uses a diffractive optic lens in combination with a dual-color focal plane array (FPA) to obtain hyperspectral images in two separate IR spectral regions. DBHSI is a passive imaging hyperspectral sensor that has been developed to the brassboard stage and successfully demonstrated in the laboratory as well as in ground-based field tests. This unique sensor collects 128 bands of images in both the midwave infrared (MWIR) and longwave infrared (LWIR) bands simultaneously. SC-700 and SECOTS are relatively small, portable hyperspectral imagers operating at the visible into the near infrared bands. SECOTS was integrated from commercial off-the-shelf components and, the data collection software was developed in-house.

Hyperspectral detection algorithms developed by the EO/IR Image Processing Branch have been applied to the hyperspectral data to detect potential obstacles to UGV navigation. The developed algorithms include the anomaly detection and target classification techniques used to detect targets of interest such as a wire (e.g. a chain-link fence) or military vehicles. In fact, we have developed three different techniques in anomaly detection—a semiparametric detector, a polarimetric detection technique, and a kernel-based nonlinear version of the well-known RX detector. The anomaly detectors detect potential targets surrounded by the background clutter without referring to prior spectral information of targets of interest. We also developed a target classification technique which, unlike the anomaly detectors, uses *a priori* target spectral

signatures in detecting materials of interest. The developed target classifier, a kernel-based nonlinear version of the spectral matched filter detector, exploits the high order correlations of spectral data thus significantly improving detection performance.

In the semiparametric detection, spectral samples are not compared as individual entities, but as individual entities and the union among these entities. An anomaly detection based on this principle provides the desirable outcome of preserving meaningful detections (e.g. a manmade objects in an open terrain), while significantly reducing the meaningless detections (e.g. transition of different regions). The semiparametric detector was derived from a solid theoretical background, and can perform remarkably well addressing very difficult anomaly detection tasks, such as detection of a virtually invisible camouflaged soldier hidden in a natural clutter background, which is invisible to a casual or critical observer of the scene. The polarimetric anomaly detector uses polarimetric hyperspectral images (horizontal and vertical polarization) produced by SECOTS to detect specular objects potentially hazardous to UGV navigation. In the kernel-based nonlinear detection, input data is implicitly mapped into a high dimensional feature space by a nonlinear mapping which is associated with a kernel function. The conventional RX and spectral matched filter expressions are then derived in the feature space which is *kernelized* in terms of the kernel functions in order to avoid explicit computation in the high dimensional feature space. The resulting *kernelized* algorithm is equivalent to a nonlinear algorithm in the original input space. The proposed nonlinear kernel-based algorithms exploit the notion that performing detection in the high dimensional feature space increases the probability of detection since they exploit the nonlinear correlations between the spectral bands. The proposed detection techniques have been applied to the SOC-700, SECOTS, and DBHSI hyperspectral images and the detection results are included in this report. In general, the proposed detectors showed significant capability in detecting potential obstacles or targets such as a chain-link fence and military vehicles.

This report is organized as follows. Extensive introductions of DBHSI and SECOTS are presented in Section 2 and 3, respectively. In Section 4 an examination of the hyperspectral data generated by DBHSI and SECOTS is presented. Target detection using a neural network detector on the hyperspectral images collected in the Office of the Secretary of Defense sponsored Collaborative Data Collection (OSD-CDC) is also presented in Section 4. In Section 5 the semiparametric anomaly detector is introduced and applied to the hyperspectral images generated by SOC-700. In Section 6 the kernel-based nonlinear detectors are derived from the corresponding conventional detectors and the detection results on the DBHSI and SECOTS hyperspectral data are included. Finally, conclusions are summarized in Section 7.

2. Dual-band hyperspectral Imager (DBHSI) and Image Processing

2.1 Overview

ARL has developed a Dual-Band Infrared Hyperspectral Imager (DBHSI) based on using a diffractive optic lens in combination with a dual-color focal plane array (FPA) to obtain hyperspectral images in two separate IR spectral regions. The DBHSI is a passive imaging hyperspectral sensor that has been developed to the brassboard stage and successfully demonstrated in the laboratory as well as in ground-based field tests. This unique sensor collects 128 bands of images each both in midwave infrared (MWIR) and longwave infrared (LWIR) simultaneously. The sensor has been developed by combining a dual-band mercury cadmium telluride (MCT) sandwich camera[1] technology from DRS Infrared Technologies, Anaheim, CA with a specially developed diffractive optic IMSS (Imaging multispectral sensing) lens from Pacific Advance Technology, Buellton, CA.[2] The image is formed in the LWIR by using the first order diffraction and in MWIR by the second order diffraction. So far this system has been used by ARL to collect target signatures outdoors for robotics applications, and by ARL/NVESD to collect target data in a NATO field test in September 2004. ARL has also used this system in the outdoor settings on our Adelphi campus. The data collection is quite straight forward and fast. ARL has successfully acquired the data using this system and processed the data as well. ARL is planning to use the prototype imager at the Dugway Proving Grounds (DPG) for a biological standoff field demo in June 2005 to evaluate the capability of this imager in biological detection.

2.2 Explanation of Technical Approach

Passive infrared remote imaging of targets requires exploitation of both the spectral signatures of the targets as well as a thermal contrast between the targets and the background scene. The basic DBIHSI technology comprises of a MWIR/LWIR focal plane array-based camera which views the far field through a diffractive lens. The image formed on the MWIR comes from the 2nd order diffraction while it comes from the 1st order diffraction for the LWIR. The motion of the lens provides the spectral resolution necessary to resolve the spectral signatures. The imaging system is shown in figure 1 with an inset of the diffractive lens.

---

[1] Goldberg, Arnold C., et al, "FOCAL PLANE ARRAYS—Comparison of HgCdTe and quantum-well infrared photodetector dual-band focal plane arrays," Opt. Eng. 42, no. 1, p. 30–46, 2003.

[2] Hinnrichs, M.; Gupta, N.; and Goldberg, A. "Dual Band (MWIR/LWIR) Hyperspectral Imager," Proc. of AIPR Workshop, p. 73–78, 2003.

Figure 1. A photograph of DBHSI with expanded view of diffractive lens in the inset.

The working principle of the DBHSI is shown in figure 2. As the lens moves it focuses different wavelengths at different points along the optical axis. There is some contribution from wavelengths other than the focused one which is removed in post processing of the image data. Also, since the lens moves in front of the FPA, it causes change in the magnification of the imaged object. This is also corrected in post processing.

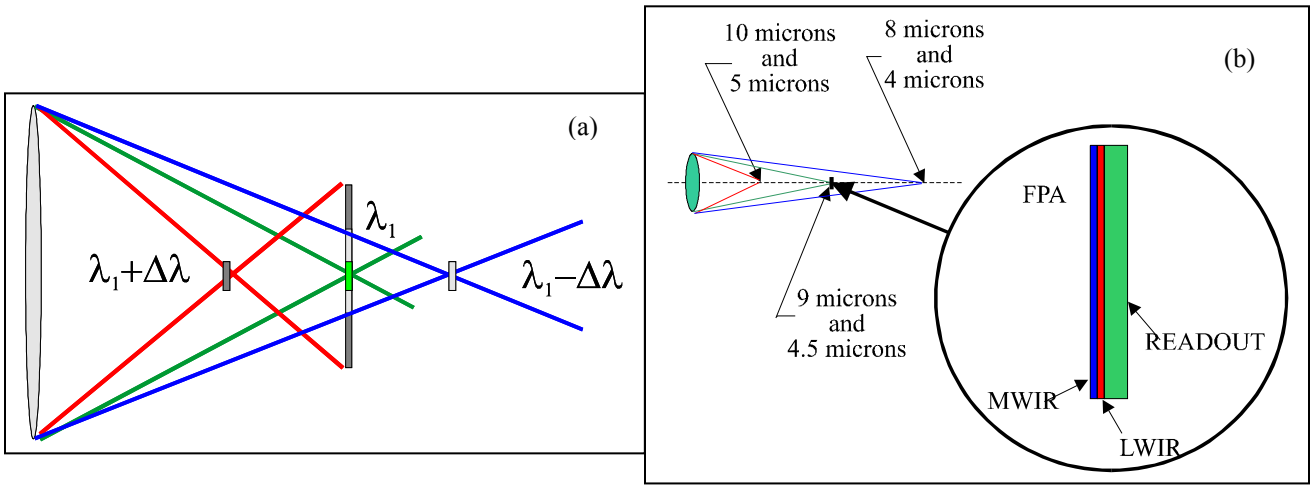


Figure 2. Operating principle of diffractive lens based hyperspectral imager is shown in (a) Single color image of full scene is obtained at one time and lens needs to be moved along horizontal direction to obtain different color images to fill image cube. When one color is focused, other colors also contribute; this is corrected in processing as well as correction for change in magnification. (b) Shows extension of this concept using a dual-color sandwich fpa using 1st and 2nd order diffraction.

When the diffractive optics lens is combined with a 2-color (LWIR/MWIR) focal plane array (FPA), it images a narrow-band in the LWIR (i.e., around 8 μm) in first order and a narrow-band in the MWIR (i.e., around 4 μm) in second order. By moving the lens over the distance required to cover the 8-10 μm spectral range, two full image cubes are obtained simultaneously: one for

8-10 μm and a second for 4-5 μm. ARL has combined a 2-color MCT FPA developed by DRS Infrared Technologies, Dallas, TX under ARL's Federated Laboratory program with a specially designed diffractive optic lens from PAT Inc., Buellton, CA to obtain simultaneous dual-color hyperspectral imaging with 128 bands in each image cube, as shown in figure 3.

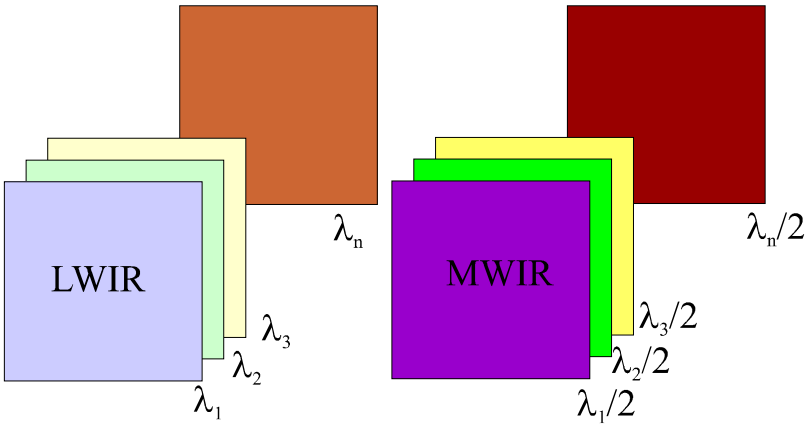


Figure 3. Simultaneous acquisition of MWIR and LWIR image cubes by SDBHSI.

2.3 Technical Specification of the System

Focal length	75 mm @9 μm Diameter 41.7 μm
f/#	3
FPA	MCT
LWIR coverage	8-10.5 μm
MWIR coverage	4-5.2 μm
Spectral resolution	0.1 μm @9 μm
Number of bands in each cube	128
Image size	320x240 pixels
Cooling	Closed cycle Sterling cooler
Power Requirements	120V
Weight	50 lbs
Collection time/scene	3 seconds

2.4 Data Collection

The Dual-Band Infrared Hyperspectral Imager is shown in figure 4. The diffractive lens and lens movement mechanism are located in the cylindrical mount on the front of the camera. The black box behind the lens contains the FPA and the Sterling cooler.

Data was taken of 17 scenes around the ARL campus. The scenes included wires, camouflaged personnel, a car, and various backgrounds. Before the scenes are imaged, non-uniformity correction is done on the camera. A midwave image cube and a longwave image cube are collected for each scene simultaneously, taking a total time of approximately 3 seconds. Each cube contains 128 different wavelengths for a total of 256 wavelengths. When the data collection for each scene is completed, the cubes from the MWIR and LWIR are combined into a single file.

As an example of the data taken, a raw image (directly out of the imager) of the car is shown in figure 5. This image is the 64th frame of the 128 frame data cubes. Notice that the imager has combined the MWIR and the LWIR data cubes placing the MWIR data on the left, and the LWIR data on the right. The 64th frame corresponds to a 9.05 micron image in the LWIR and a 4.68 micron image in the MWIR. Visible photographs were also taken of the scenes to provide ground truth. The ground truth image of the car is shown in figure 6.

Figure 4. Dual-Band Hyperspectral Imager.

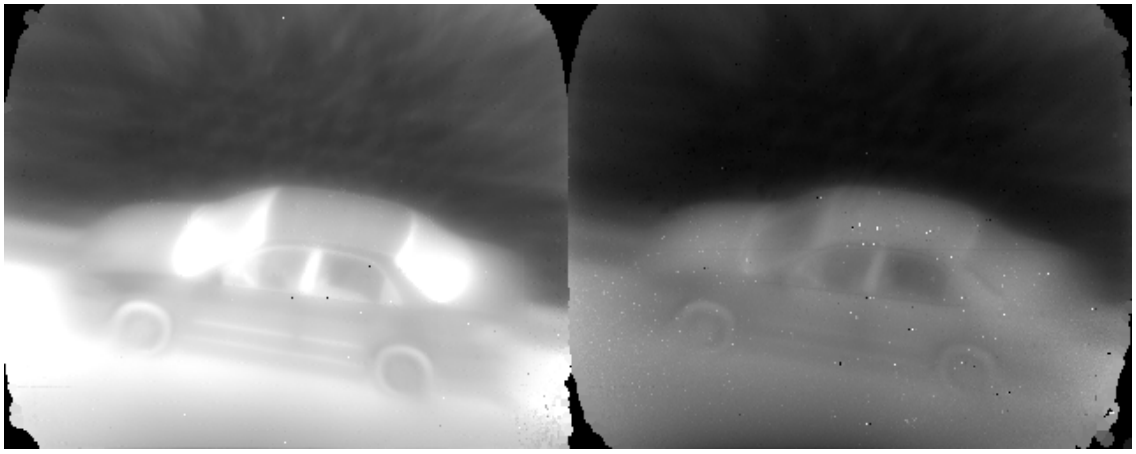


Figure 5. Raw data directly from the camera. Frame 64.

Figure 6. Visible ground truth image of the car.

2.5 Noise Issues

The raw images produced by the camera are blurred due to contributions from wavelengths that are out of focus, as shown in the sample spectral image in figure 5. The algorithm used to get rid of this blur is called spatial deconvolution. After the data of figure 5 was collected, the LWIR and MWIR data cubes were separated. Spatial deconvolution was then performed on the images, and this post-processing did indeed sharpen the image. However, large spots appeared in the images, as shown in figures 7a and 7b. These spots appeared as small black and white spots in figure 5, and are due to dead pixels and bad pixels on the FPA.

Automatic target recognition (ATR) algorithms were performed on these processed images in order to see if they could be used to identify certain features in the images. The ATR algorithm finds features of interest by searching for anomalies. These spots were seen as anomalies by the ATR algorithm, and then mistakenly seen as features of interest. If information is to be extracted from these images using ATR, the dark and bright bad pixels in the images would have to be removed prior to the deconvolution process, which seems to amplify these.

Upon closer inspection of the raw data, it was seen that the large spots were produced from noisy pixels during the spatial deconvolution process. In order to reduce the large white and black spots, the noise must be reduced. Even though a procedure is employed to reduce the noise during the initial calibration procedure, the images from the camera were still too noisy. We needed an additional tool to reduce the noise after the images were produced by the camera.

The first attempt at reducing the noise of the raw images was to use a median filter. After the MWIR and LWIR cubes were separated, a median filter was used on the cubes. Using the median filter did indeed reduce the noise. The spatial deconvolution was performed on the images filtered through a median filter, and the result showed no large spots, as shown in figures

8a and 8b. There existed another problem however. It appears that the use of the median filter changes the spectral content of the images, giving rise to false spectral characteristics in the data.

In order to fix this problem we took a second approach. A MATLAB program entitled replacebadpixels.m was written to take the raw images from the camera, identify the bad pixels, and replace the bad pixels with values that better resembles the actual data. The program is described in the next section.

Figure 7a. MWIR data that has undergone spatial deconvolution. Frame 64.

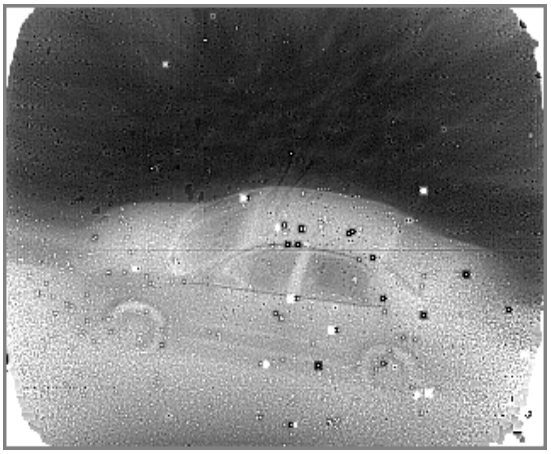


Figure 7b. LWIR data that has undergone spatial deconvolution. Frame 64.

Figure 8a. MWIR data processed by a median filter and then spatial deconvolution. Frame 64.

Figure 8b. LWIR data processed by a median filter and then spatial deconvolution. Frame 64.

2.6 Replacebadpixels.m

Replacebadpixels.m is a MATLAB function designed to take an image data cube, search for bad pixels, replace these bad pixels with a value that more closely approximates the true data value, and write a new file (new image data cube). The output of the function is the new image data cube, as well as two matrices which locate where the bad pixels occurred.

There are two basic types of bad pixels. The first are pixels which blink on and off as the image cube is scanned from one wavelength to another. In other words, the data looks good as the wavelengths are scanned until suddenly the pixel gives a false peak or valley at one particular wavelength. The data then goes back to good data as more wavelengths are scanned. Some pixels blink many times while others blink only once. The second type of bad pixel is one that is constantly on or constantly off.

After reading in the image data cube, replacebadpixels.m finds the first kind of bad pixels. It does this by looking at one pixel at a time, and scanning through the wavelengths. The value of the pixel at one wavelength is compared with the value of the pixel at the next wavelength. If the value at the next wavelength is higher (or lower) than the present one by some user defined percentage, the pixel is flagged as a bad pixel. The program does this for all the pixels in the image. Once a pixel is flagged as bad, it is considered bad for all the wavelengths. It is important to note that at this stage the bad pixels are just flagged as bad pixels. Nothing is done at this point to replace the bad pixels with another value.

The next step in the program is to find the second kind of bad pixels (those constantly on or off). This is done by looking at one pixel and comparing its value to the median of its eight nearest neighbors (all 9 pixels are in the same wavelength frame). If the value of the pixel of interest is higher (or lower) than the median by a user defined percentage, then the pixel is flagged as a *possible* bad pixel. At this point it is not known if the pixel is indeed bad, or if the pixel represents real data. The program then compares the value of the pixel with its nearest neighbors. If there are a least two other neighbors with similar values as the pixel, the pixel is not considered bad. Otherwise, the pixel is considered bad. The process is repeated for all the pixels in that wavelength, and then the next wavelength is tested. Once a pixel is flagged as bad, it is considered bad for the whole data cube. Again, only the pixel location is flagged. Nothing is done during the search to replace the bad pixels.

At this point in the program, all the bad pixels are located. The program then starts at the first wavelength and goes to the locations of all the bad pixels. At each location, the median of its nearest neighbors is calculated and stored for later use. The pixels are not replaced until all the medians are found for all the bad pixels. Once all the medians are found, the bad pixel values are replaced by the medians. Then the program goes on to the next wavelength, and the process is repeated for all wavelengths. During the final step of the program, the new data cube is written to a new file.

Frame 64 from the new data cube is shown in figure 9. Comparing figure 9 with figure 5 it can be seen that the new data cube is less noisy than the raw data directly out of the camera. The program was successful in reducing the noise. In this example, the user defined thresholds used to find the first kind of bad pixels was 8% and the threshold used to find the second kind was 15%. These were found to be fairly high thresholds, resulting in numerous bad pixels remaining in the new data cube. The number of bad pixels of the first and second kind was found to be 488 and 446 respectively. There were 272 of these pixels that were identified as belonging to both types. Subtracting this redundancy leaves a total number of 662 bad pixels (0.4% of the total number of pixels). The program can be run with continually lower thresholds, resulting in more of the bad pixels being identified. Section 7 will show an example using lower thresholds.

Spatial deconvolution was performed on the new data cube (after separation was performed), and the result can be seen in figures 10a and 10b. Comparing figures 7a and 7b with figures 10a and 10b, it can be seen that the processed new data cube is devoid of the large bright and dark spots that caused problems in the ATR algorithm. There are some smaller spots in the processed new data cube that may still cause some problems. This still needs to be tested. If problems still occur, a lower threshold can be used in the program to reduce the number of smaller spots.

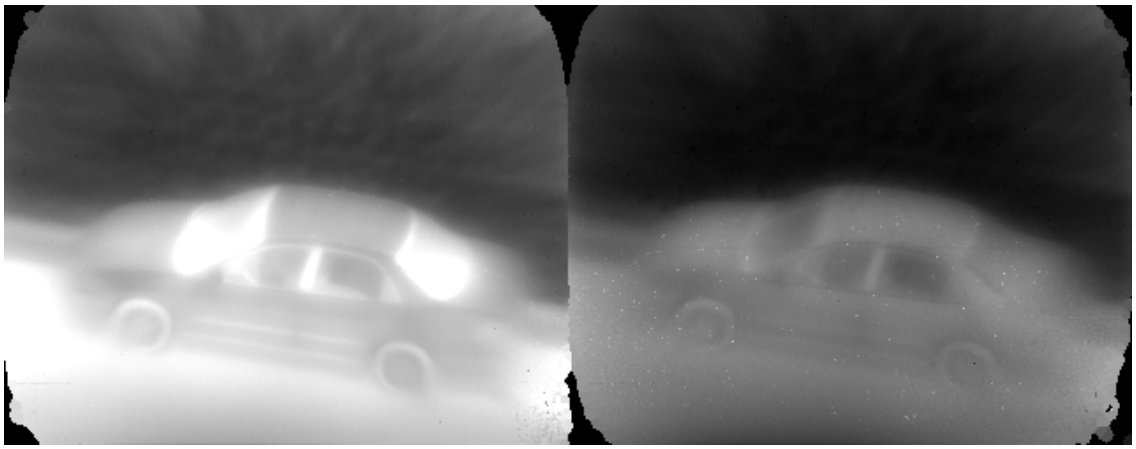


Figure 9. New data written by the replacebadpixels.m function. Frame 64.

Figure 10a. MWIR new data after spatial deconvolution. Frame 64.

Figure 10b. LWIR new data after spatial deconvolution. Frame 64.

2.7 Lower Threshold Example

The previous section showed an example of a data cube processed by replacebadpixels.m using high thresholds. The thresholds can be lowered resulting in more of the bad pixels being identified and fixed. As an example of lowering the thresholds, the raw car data of figure 5 was processed using thresholds of 1.5% (for both types of bad pixels), and the result (called the 1.5 data cube) is shown in figure 11. Comparing figure 9 with figure 11 shows an obvious reduction in noisy pixels with the lower thresholds. Lowering the thresholds resulted in the identification of 2,507 bad pixels of the first kind and 2,034 bad pixels of the second kind. There were 880 pixels that were identified as bad pixels of the first kind that were also identified as bad pixels of the second kind. Subtracting this redundancy gives a total number of 3,661 pixels that were identified as bad. This is 2.4% of the total number of pixels.

At this point it is important to note a major difference between the two methods used to deal with the noisy pixels. The median filter approach affects all of the pixels in the data cube, and affects them in a way that results in corrupted data for all the pixels. On the other hand, only a few percent of the pixels are altered using replacebadpixels.m. Therefore, the majority of the pixels do not have their spectral content altered at all. It is true that the few percent do have their spectral content altered, but they were bad pixels to begin with. They never recorded an accurate spectral content. What has been done using replacebadpixels.m is to replace the bad pixels with a value that approximates what the pixel would have recorded had it worked.

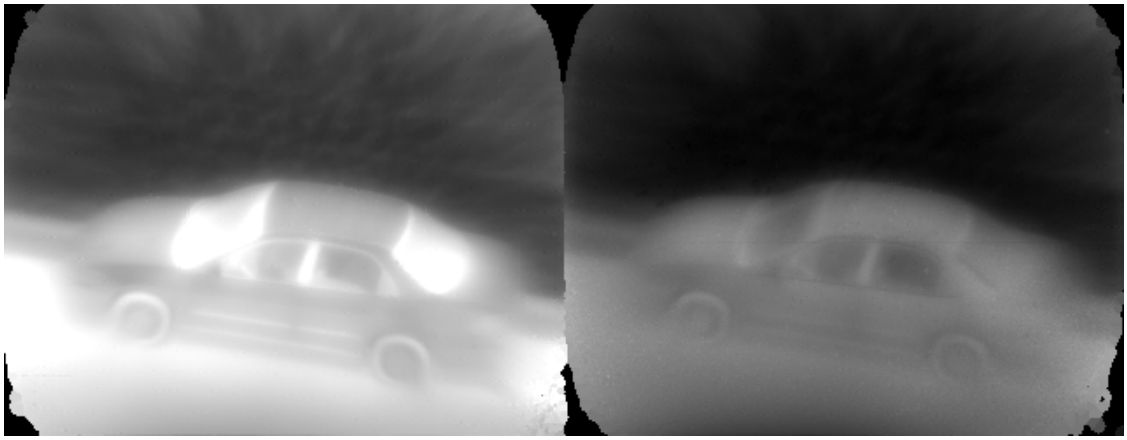


Figure 11. 1.5 data cube produced using thresholds of 1.5%. Frame 64.

2.8 Final Processing Steps and Results

The raw data cube that is produced by the camera must undergo processing steps in order to extract the spectral information it contains. The first step is to identify the bad pixels in the images. This is done using replacebadpixels.m, and using low thresholds. The second step is to divide the data cube into the MWIR data cube and the LWIR data cube. The raw data from the camera had the two cubes fused together, and they need to be separated before any other processing can be done. Once the data cubes are separated, each undergoes spatial deconvolution in order to remove the blur caused by out-of-focus wavelengths.

The final processing step is to compensate for the differences in magnification between the different wavelengths. Due to the dispersive nature of the diffractive lens, each wavelength has a different magnification associated with it. This results in objects in the scene changing in size and position in the image as the wavelengths are scanned. Magnification compensation is performed on the data cubes so that every wavelength has the same magnification, the object size remains fixed, and the pixel location of the object is the same for all wavelengths.

An example of a hyperspectral data cube that has undergone all of the processing steps is shown in figures 12–17. All six figures began as the data cube of figure 5. They all underwent bad pixel replacement using thresholds of 1.5% for both types of bad pixels. They all underwent separation into separate cubes, spatial deconvolution, and magnification compensation. Figures 12, 14, and 16 show the MWIR data cube at wavelengths of 4.16 microns, 4.65 microns, and 5.15 microns respectively. Figures 13, 15, and 17 show the LWIR data cube at wavelengths of 8.01 microns, 9.00 microns, and 10.01 microns respectively. Now that this data has been collected and the data cubes processed, the spectrum of any object in the field of view can be extracted. Figure 18 shows the LWIR spectrum of a point on the roof of the car. Figure 19 shows the MWIR spectrum of that same point on the car roof.

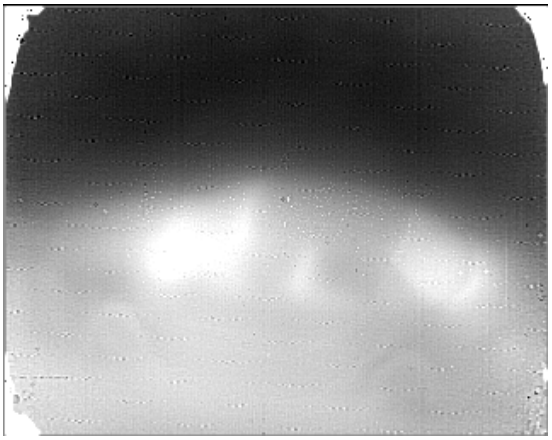


Figure 12. MWIR image at 4.16 microns.

Figure 13. LWIR image at 8.01 microns.

Figure 14. MWIR image at 4.65 microns.

Figure 15. LWIR image at 9.00 microns.

Figure 16. MWIR image at 5.15 microns.

Figure 17. LWIR image at 10.01 microns.

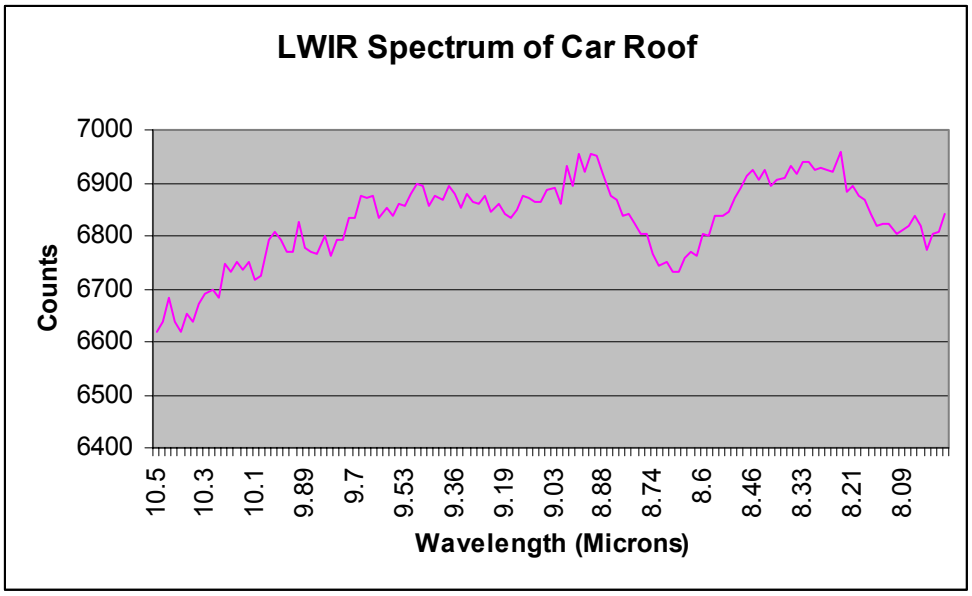


Figure 18. LWIR spectrum of a point on the car roof.

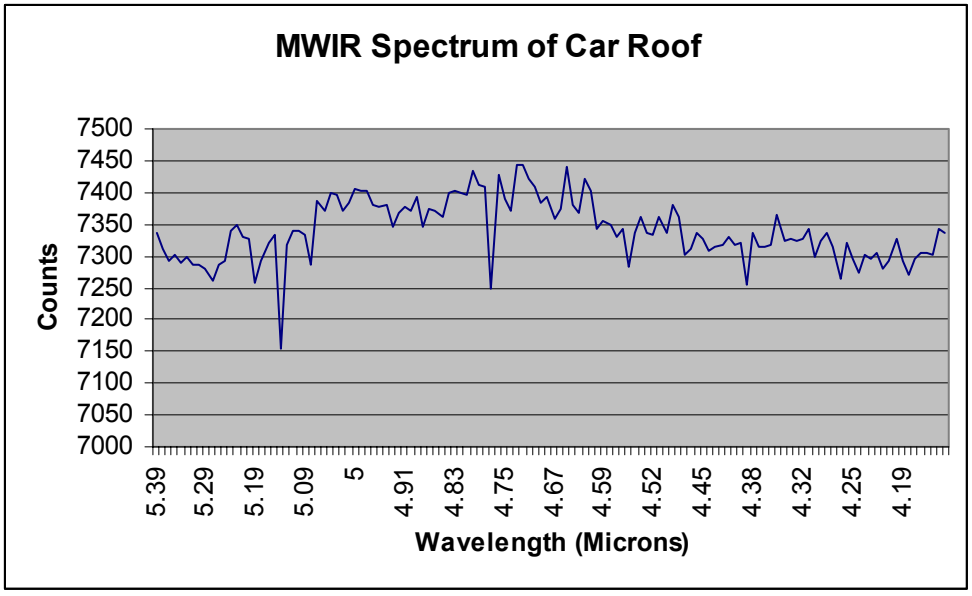


Figure 19. MWIR spectrum of a point on the car roof.

2.9 Future Effort

The present imager can be further reduced in size and weight. The reason for its bulky size and weight is that it is a hybrid prototype system which was put together by ARL by combining the camera drive electronics and data acquisition boards from the SE-IR Corp., Goleta, CA with the PAT lens and software. An entire system can be designed that will be more compact through the use of FPGAs in the electronics and by improvements in packaging.

3. Acousto-Optical Tunable Filter Image Collection System (SECOTS)

3.1 Overview of the SECOTS system

In support of robotics applications, the EO/IR Image Processing Branch has collected hyperspectral imagery using an Acousto-Optical Tunable Filter (AOTF) image collection system, called SECOTS. This system (shown in fig. 20) is capable of collecting up to 80 user defined bands from the visible into the near infrared. It was integrated from commercial off-the-shelf components and the data collection software was developed in-house. SECOTS has collected multi-season imagery at outdoor locations within the ARL Adelphi, and NIST Gaithersburg campuses. With a total system weight of approximately 50 lbs, SECOTS is relatively easy to deploy, and has been successfully mounted to a generator-equipped HMMWV and driven off-road at low speeds during a data collection at NIST. It employs an Acousto-Optical filter supplied by the Brimrose Corporation, attached to a sensitive monochrome CCD from Qimaging. Digital images are captured from the Qimaging CCD to a portable computer via its Firewire (IEEE 1394) output.

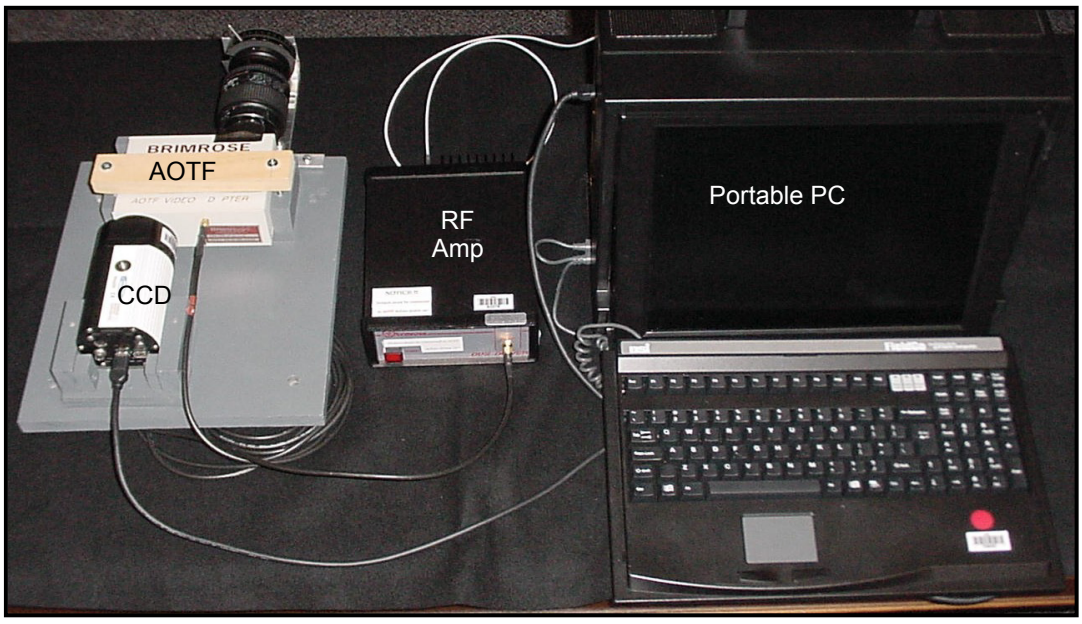


Figure 20. AOTF image collection system.

3.2 Principal of Operation

An AOTF acts as an electronically tunable spectral bandpass filter. Its operation is based on the acoustic diffraction of light in an anisotropic medium. A radio frequency (RF) signal is amplified and applied to a piezoelectric transducer which is bonded to a birefringent crystal. In the SECOTS system, a Tellurium Dioxide crystal is used.

The transducer generates acoustic waves in the medium, and the propagating waves create a periodic modulation in the index of refraction. This modulation results in a moving phase grating which will diffract portions of the incident light beam under the proper conditions. The incident light is separated into two diffracted monochromatic beams and two undiffracted beams as shown in figure 21. The two diffracted waves are orthogonally polarized, and the wavelength (nm) of the diffracted waves is a function of the frequency (MHz) of the applied RF signal. By varying the frequency of the applied RF signal, one can select or "tune" the wavelength of the diffracted waves.

In the SECOTS AOTF, a beam stop is placed so that it will block the vertically polarized "tuned" beam and the two undiffracted beams if the incoming light falls within the acceptance angle. Under these conditions, only the horizontally polarized "tuned" beam will reach the imaging CCD.

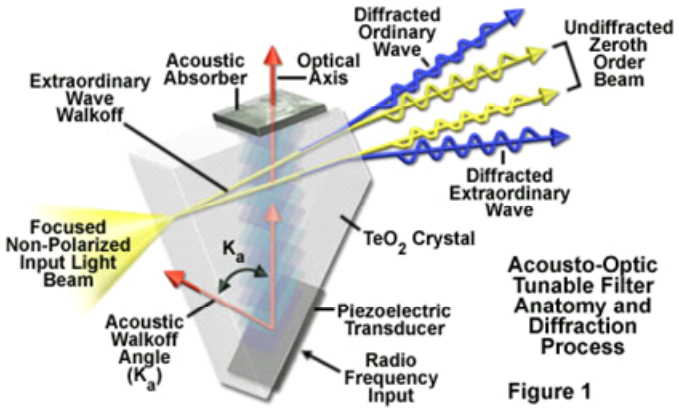


Figure 21. AOTF anatomy and diffraction process.

3.3 Technical Specifications of the System

A. Brimrose AOTF Video Adaptor Specifications:

Material	TeO_2
Active Aperture	10 x 10 (sq-mm)
Spectral Range	500-1000 nm
Spectral Resolution	5nm @ 833nm
Spatial Resolutions (TV Lines)	1000 x 1400 @ 450nm
Connection to CCD	C-mount (w/ adaptor)
Connection to Front Objective Lens	Nikon Mount
RF Drive Power	2 W
RF Connector	SMA
Drive Frequency	86-199 MHz
Polarization	Horizontal
Weight	0.5 kg
Dimensions	184.2 x 148.5 x 54.0 (mm)

B. Qimaging Retiga EX CCD Camera Specifications:

Sensor Type	ICX-285 progressive scan interline Monochrome CCD
Pixel Resolution	1360 horizontal x 1036 vertical
Pixel Size	6.45 um x 6.45 um
Format	2 / 3 ”
Quantum Efficiency @ 400 nm	45%
Quantum Efficiency @ 500 nm	70%
Quantum Efficiency @ 600 nm	67%
Digital Output	10-bit to 12-bit
Digital Interface	IEEE1394
Signal-to-Noise Ratio (SNR)	60dB
Power Requirements (uncooled)	6W
Power Requirements (cooled)	11W

C. SECOTS Overall Specifications:

Polarization Diversity:

A half-wave plate phase retarder is added to the front of the AOTF in the SECOTS system to provide the ability to vary the phase of the linear polarization recorded by the CCD. By rotating the phase retarder to an angle *theta*, the orientation of the polarization will be “rotated” by 2**theta*. For most data collections, four polarization orientations are collected: 0, 90, + 45, and -45 degrees.

Spectral Range:

	Spectral Range	# 5nm bands	Illumination
Brimrose AOTF	500-1000 nm	100	Bright lab sources
SECOTS End-to-End	540-840 nm	60	Normal outdoor illumination

The Brimrose AOTF can operate over a spectral range of 500 to 1000 nm, using bright laboratory light sources. The useful spectral range of the SECOTS system, using normal outdoor day-time illumination, was found to be approximately 540 to 840nm, providing 60 bands in 5nm increments.

Data frame rate:

	Time to process 1 band	Time to process 60 bands	Illumination Source
Brimrose AOTF max	250 microseconds	15 milliseconds	Laser
Qimaging CCD max	.20 seconds	12 seconds	Bright artificial
SECOTS maximum	.5 seconds	30 seconds	Bright outdoor
SECOTS typical	1-2 seconds	1-2 minutes	Typical outdoor

The SECOTS end-to-end frame rate is primarily determined by the CCD exposure time required to produce a low-noise image, using normal outdoor natural illumination. The typical time required to collect each frame (or band) with the SECOTS system is on the order of 1 second, with the complete 60 bands requiring approximately 1 minute. Given the relatively slow rate of collection, SECOTS cannot be used when the sensor or objects of interest are moving, or when the illumination in the scene changes greatly within 1 minute.

3.4 Data Collection

For this study, three sets of images were collected (as shown fig. 22) in support of the hyperspectral detection of wires and fences, for robotic navigation. For each image set, 240 images were collected, 60 bands x 4 polarizations.

	Image Set #1	Image Set #2	Image Set #3
Sensor Location	balcony of Zahl bldg	sidewalk near K-Lot	sidewalk near K-Lot
Range to scene	34 M	19 M	19 M
Scene Description	rising slope with cut grass and bare soil	uncut thicker vegetation and dead leaves	uncut thicker vegetation and dead leaves
Illumination	brightly lit with full sunlight	cloudy day, variable lighting filtered by trees & clouds	sunny day, variable lighting filtered by trees & clouds
Man-made Objects	unweathered horizontal suspended wire between stakes and chain link gate	weathered horizontal barbed wire on top of weathered chain link fence	weathered horizontal barbed wire on top of weathered chain link fence

Figure 22. Scene and Sensor Setup of Image Set #3.

3.5 Image Registration Issues

A comparison of the 540 nm image to the 840 nm image in a 60 band sequence shows the entire image translates 13 to16 pixels in the X-direction and several pixels in the Y-direction. This type of image shift (shown in fig. 23) is typical in AOTF systems, such as SECOTS, when a small optical wedge has not been included to compensate for this effect. As collected, the image sets could not be used for hyperspectral processing, due to the mis-registration of spectral information for each pixel through the 60 bands.

A commercial software package, Image Pro Plus, from Media Cybernetics was used to manually translate each image in the image set for a "best match" with the 540 nm image. Finally, Matlab was used to crop all the images to remove for the "blank areas" produced by the first step. The resulting registered image sets have the following image sizes:

	Collected Image	Registered Image
Image Set #1	1360 x 1036	1344 x 1033
Image Set #2	1360 x 1036	1346 x 1033
Image Set #3	1360 x 1036	1345 x 1033

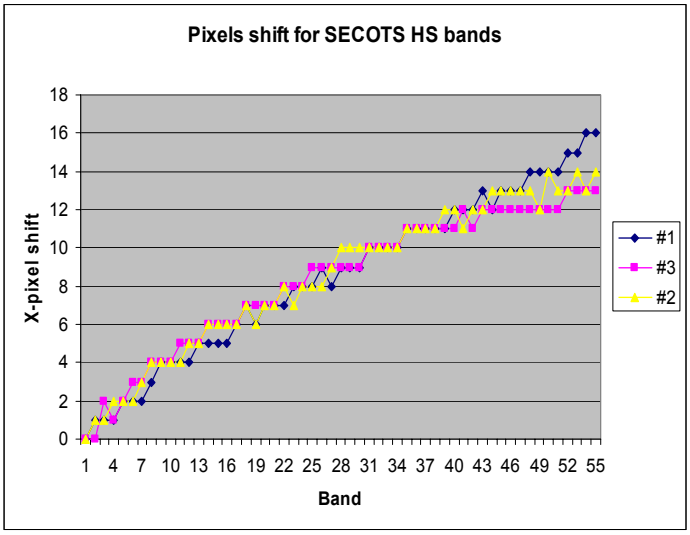


Figure 23. Pixel shift for SECOTS HS bands.

3.6 Illumination Variation Issues

In addition to the image registration issue, the spectral information through all 60 bands will be effected by variations in illumination that occurred during the 1 minute time span it took to collect the images. Similarly, there is a gap in time of at least 1 minute or more for the same band at different polarizations.

For Image Set #1, the variation in illumination was minimal because of the steady conditions that day. For Image Sets #2 & 3, there was sufficient variation in illumination to make it difficult to compare spectrally the images of the same wavelength between the 4 polarization settings. Within a polarization setting, the data was collected to minimize the variation of illumination over the 60 wavelengths.

3.7 Future Effort

Several modifications to the SECOTS system could make it a more effective data collection system for hyperspectral processing. An optical wedge could be integrated into the Brimrose AOTF, to minimize the image drift. At the same time, an upgrade to the Image Pro Plus software package would automate the manual image registration and greatly speed up that process.

Higher collection rates would help to reduce effects of time varying illumination. The SECOTS frame rate could be increased by using a CCD with less resolution (more energy per pixel) than the Qimaging CCD. The CCD would require shorter exposure times, but it would decrease the effective detection range for thin objects such as wires. In addition, a CCD with a standard analog video output could increase the SECOTS frame rate up to 30 frames per second. A video frame grabber with a Software Development Kit (SDK) would be needed, and the data collection software would have to be re-written to control the video frame grabber.

4. Examination of Available Hyperspectral Data and Using Neural Network with OSD Data

4.1 Examination of Data from DBHSI

The specification for the file format of the DBHSI sensor data was not supplied. The only known information was the HSI cube dimensions: 640 by 240 by 128(bands) and the actual file size (19,662,080 bytes). The data received had been separated into two separate files (MWIR and LWIR) so that the cube dimension of the data was 320 by 240 by 128. Dividing the file size by the dimensions reveals the data value size.

19,662,080/ (320*240*128) =2.0001202

So the data likely consists of 2-byte values. The fact that is not an integer value indicates there is extra data, most likely as a header, a footer, or both. Subtracting the dimension size showed:

19,662,080-(320*240*128) =1280

bytes remaining.

Using ENVI, the HSI file was opened using the above dimensions. The 2-byte factor indicates 16 bit values of some sort, most likely unsigned 16 bit integers (as negative radiance values are unlikely). Initially, the value of the offset was set to 1280. Knowing that the sensor collects one band at a time indicated that the data was most likely stored in BSQ (band sequential format). Viewing the results on the display, it was obvious that the header size (offset) was incorrect. The hyperspectral data file was opened with the ENVI 'data viewer' to look at the raw values. The HSI data appeared to start at byte 256. Changing the header size to 256 and displaying the first band again, the correct image was displayed. Obviously, the PAT software places a header and footer around the cube data.

With the HS data file loaded into ENVI, we used the 'animate bands' feature to run through all bands and display them as a movie on the screen. Upon observing the movie, it was immediately noticed that the 'bad pixels' were moving radially toward the center of the image. This was caused by the magnification correction. Unfortunately, this made all 'pixels' traversed by the bad pixels' spectrally inconsistent.

It was also noticed that there was some jitter in the image, most noticeably at the edge of objects. This would cause problems for spectral processing, as some of the bands of a pixel's spectra will be from one spatial location and other bands will be from different spatial locations. This observation also implies any movement in the scene during collection of the entire HS cube will generate corrupted spectral samples. It was also noticed that the expected black body curve was

not seen in the LWIR data. This was later explained by the hardware group. This further complicates the preprocessing that must be done to the data collected by this sensor, before it can be regarded spectral data that can be spectrally processed.

4.2 Examination of the data from the SECOTS sensor

This visible wavelength hyperspectral data also suffers from jitter because it also collects only on band at a time. There is also a somewhat uniform drift of the scene across the image, from one band to the next. This was addressed in the SECOTS section.

4.3 Use of OSD-CDC Data for Anomaly Detection and Material Identification

In May/June 2004, members of the EO/IR Image Processing Branch participated in the Office of the Secretary of Defense sponsored Collaborative Data Collection (OSD-CDC) at Ft. Hunter Liggett, in support of a visible hyperspectral sensor from ARDEC-Picatinny Arsenal. They have a modified SOC 700 Vis-NIR hyperspectral imager. Over a two week period, we collected over 125 GB of spectral data. In addition, 6 GB of high resolution digital color camera ground truth imagery was collected using a 5 mega-pixel Nikon CoolPix camera. Four of the 'official' targets were commercial vehicles (2 SUV's and 2 Pick-up trucks). Towards the end of the collection, we were given operational control of the commercial vehicles and their drivers, for a day. During this time, we collected 194 spectral cubes of the commercial vehicles, at different locations on the test-range. We also had the drivers out of the vehicles and positioned in different locations (bright sun, in shadows, under trees and so on) and aspects, at the various test locations. The individuals usually had man-portable weapon tubes (actual containers of expended munitions: Stinger manpad, AT-4 and RPG launcher).

Twelve of these cubes had individuals close to the sensor, 2 to 20 meters from the sensor. From these frames, 22 different classes of materials were identified and samples were extracted to form a training set for neural network (NN) material classification tasks, see table 1 for a list of classes and numbers of pixels extracted. The data sampling was done by hand using ENVI. An effort was made to collect 100 sample pixels of each class from each image, from diverse areas in the HSI not just in a local area. Of special note is the fact that the 'stinger-yellow' class was only visible as two small circles on the tube and visible in only one of the images used for spectral sample extraction, resulting in only 16 pixel samples.

Appendix A contains plots of the spectra of each class, taken from 16 example pixels across the training class set. This shows the variations possible even in a relatively well lit setting.

Appendix B contains plots of the samples in Appendix A, normalized and ready for presentation to a neural network. The spectra are normalized into a range of -1 to 1 for two reasons. First, it is part of an investigation to get a more generalized NN and faster training of the network by taking the intensity out of the data. Hopefully, this will allow the NN to ignore the variations in intensity. Secondly, it also allows the network to train faster by allowing the network to know ahead of time what the maximum and minimum expected values are for the data.

Table 1. List of training classes and details.

#	Class Name	Sample location	**# of samples**
1	Tree leaves	sampled from many trees in each frame	**1200 pixels**
2	Tree bark	sampled from many trees in each frame	**1200 pixels**
3	grass – brown	sampled from many regions in each frame	**1200 pixels**
4	grass – yellow	sampled from many regions in each frame	**1200 pixels**
5	grass – gray	sampled from many regions in each frame	**1200 pixels**
6	grass – green	sampled from many regions in each frame	**1120 pixels**
7	dirt	sampled from one area where ground cover cleared away	**600 pixels**
8	macadam	road surface in scene	**800 pixels**
9	skin from person 2	sampled from face and arms/hands	**900 pixels**
10	hat – green	From hat on person 2	**800 pixels**
11	hat – blue	From hat on person 1	**300 pixels**
12	pants – dark	From pants of person 1, close to sensor	**300 pixels**
13	shirt – dark	From shirt of person 1, close to sensor	**300 pixels**
14	blue jeans	From pants of person 2, close to sensor	**800 pixels**
15	shirt – blue	From shirt of person 2, close to sensor	**850 pixels**
16	stinger tube	sampled from tube at various aspects	**300 pixels**
17	stinger yellow dot	From 2 yellow dots at front of tube in one frame only.	**16 pixels**
18	boots – brown	Sampled from person 2. Feet visible in only one frame.	**100 pixels**
19	black vinyl	sampled from glass case on person 2's belt	**530 pixels**
20	pick-up white	sampled from body panels of pick-up	**600 pixels**
21	pick-up blue	sampled from body panels of pick-up	**300 pixels**
22	Skin from person 1	Sampled from face and hands	**300 pixels**

Several architectural variations on the simple three layer back-propagation of error neural networks were and are being trained with portions of this data. Class 22 is not being used to train networks at this time, as we are interested if a network, trained with skin samples from one person, can correctly identify the other. Current evidence indicates it can (see Figures 24–26).Empirical evidence from previous experiences using simple back-propagation networks indicates that the ratio of the number of examples of individual training classes should be less than 3 to 1, but need not be less than 2 to 1, with minimal loss of performance. Thus for this work, a maximum of 800 samples will be used. For classes with less than 800 examples, the training data will be replicated in its entirety to bring the total number of examples for the class to between 600 and 800 examples. This approach, replicating training sets to keep the ratios between 2 to 1 and 3 to 1, has worked well in the past.

For example, class 7 had 600 examples so no adjustment will be made to this class. Class 11 has 300 examples so it will be replicated once for a total of 600 examples. Class 17 only has 16 examples so it is replicated 50 times for 800 examples. If this is not done, the network would find that by ignoring the 16 values (returning anything) it could quickly minimize the mean squared error (MSE) of the entire network. Ensuring a somewhat equal number of training examples, ensures no training example has too much or too little impact on the MSE of the entire network, thus leading to a better overall solution.

4.4 Results

A 21 class problem (material identification) has shown promising results, as well as some troubling misidentifications. Figure 24 shows the high resolution digital image, showing person 1 near the sensor.

Figure 24. Truth image for Cube_117.

Figure 25 shows a single band of the hyperspectral image cube. Figure 26 shows the output of node 9 (the skin-2 class) of neural net trained to classifiy the 21 classes individually. It is a simple back-probagation of error neural network. This result shows that the network, which was trainined with skin spectra from person 2, can in fact identify the skin of person 1.

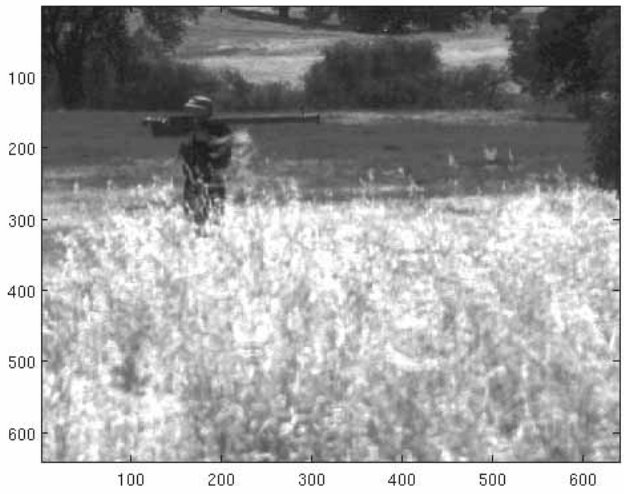


Figure 25. Band 55 of Cube_117.

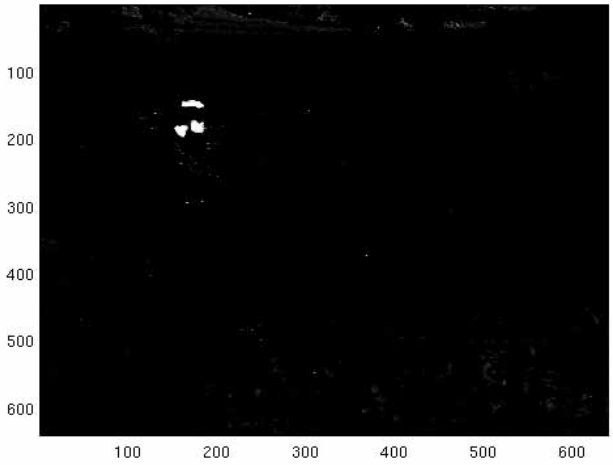


Figure 26. Output of NN layer trained to highlight skin-2. This person is Person 1, not Person 2. (Note: it shows the hands and exposed face very well.)

Unfortunately under different circumstances, the NN can make a significant error. Figure 27 shows a classification map (for the image shown in fig. 28) built using the NN output for the 21 class problem. To generate this image, each output class of the NN is examined and the largest value wins, a winner take all approach. As we can see the network had highlighted a significant portion of the pickup truck on the right as class 'skin-2,' which is obviously wrong, figures 27 and 29.

Examining figures 27 and 30, we also note that the confusion between classes appears most

prominent in the areas of shadow at the base of the trees. This is interesting and significant because the neural network was not trained with any data taken from areas of shadow. Extracting additional training data for the various classes that are in shadow may allow the network to appropriately deal with the shading. However this problem may be more serious, as this sensor is a visible near-IR system; it is highly dependent on the illumination of the scene. Areas in shade are receiving different illumination and therefore may require each material to be further broken down in to 'in sunlight' and "in shade" classes for proper classification.

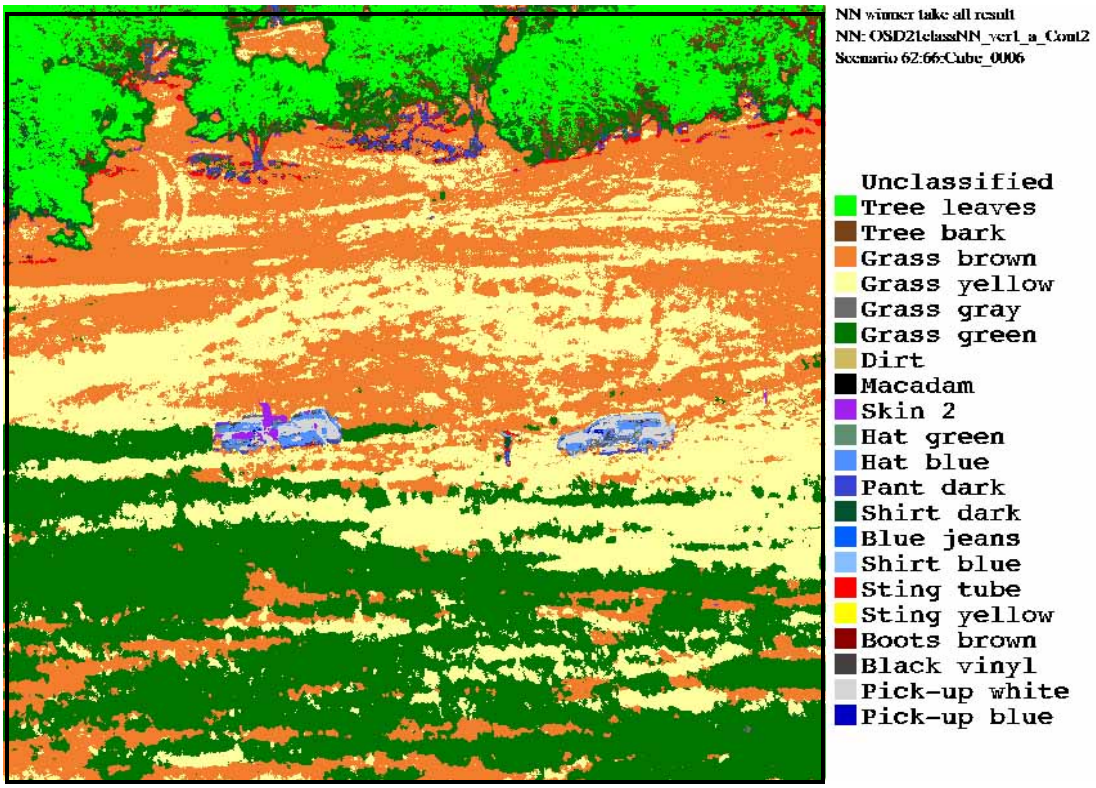


Figure 27. Winner take all output image from a neural network trained to identify the 21 classes.

Figure 28. High resolution digital image of vehicle deployment matching Figure 29. (Note: this is only an image for positional details of the vehicles and for truthing the natural materials in the HS image.)

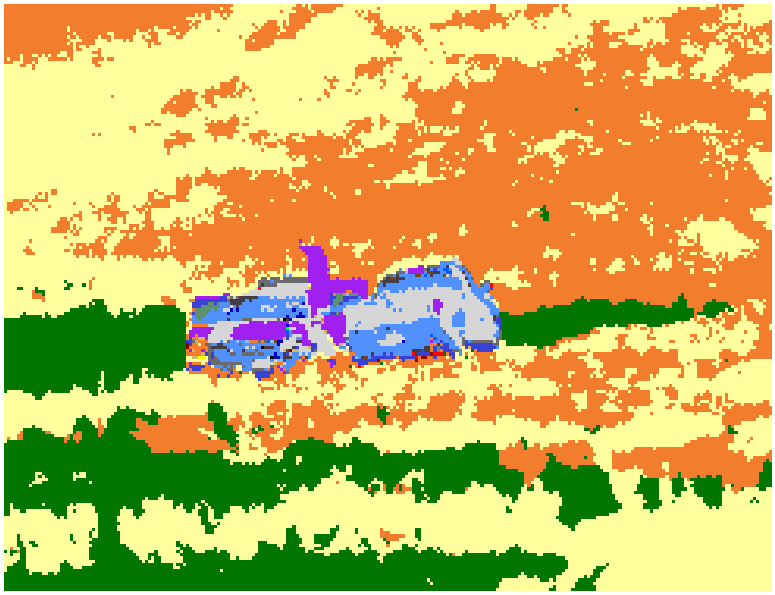


Figure 29. Blow-up of trucks on right. Note classification error of truck as class Skin-2 (purple).

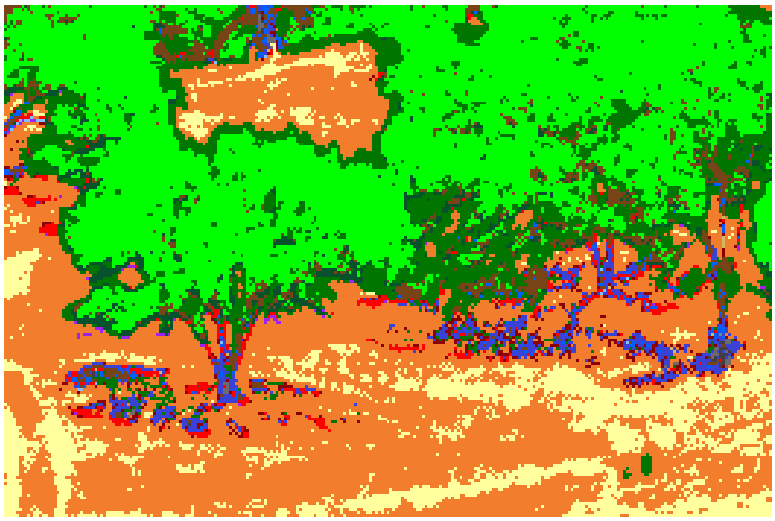


Figure 30. Note similar errors with Class 9 (skin-2; purple) and Class 16 (stinger-tube; red).

Another way to approach this problem with a neural network, is to attempt to train the network to just separate the spectra into two classes: Natural and Man-made. To do this only two output nodes are needed, one for the Natural class and one for the Man-made class. The training data is collected, classified as one of the two classes, and the network is trained. Figure 31 shows an output surface from the neural net trained to do the natural vs. man-made separation, for the same hyperspectral cube that was processed to generate figure 27. It shows the output, black indicating a strong natural responsese and and white a strong man-made response. This looks very good, but of course this is an easy image.

Figure 32 shows a much more difficult image; an SUV and man in heavy shade and against a tree-line. Figure 33 shows the output of the same Natural vs Man-made network. The results are not too bad. While we do see the vehicle and man highlighted there are also many false alarm pixels in the scene. Again we see trouble with shadowed pixels. Note the horizontal line across the image about 2/3rds down the image, is a macadam road, which is being highlighted correctly.

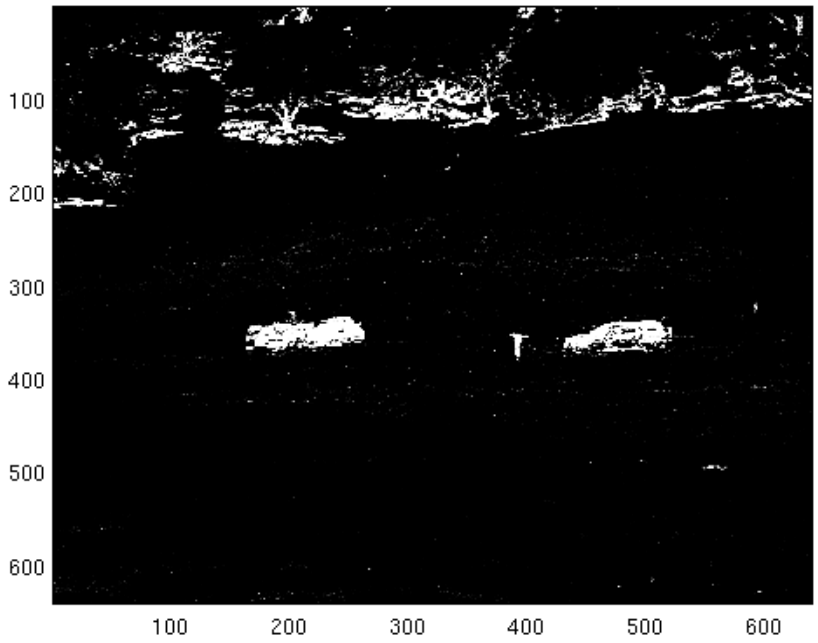


Figure 31. Output of a NN trained to do Natural vs. Man-made separation. White is high confidence is man-made.

Figure 32. High resolution color digital image of a scene with the SUV and man in shadow and against treeline.

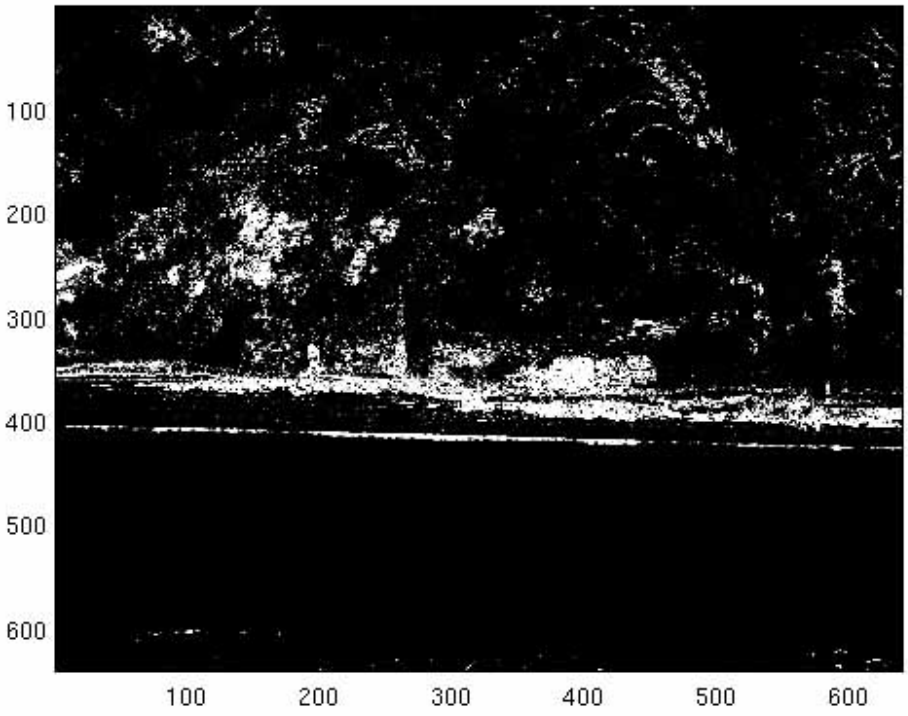


Figure 33. Results image from the Nature vs. Man-made trained neural network.

4.5 Band Selection

We also are investigating what NN's can tell us about the prospect for band selection and how the number of classes may impact the number of bands needed for anomaly detection or material identification. Neural networks were trained to do every unique 2 class combination from the 21 different classes, using 1 hidden node, a total of 210 different networks. The networks had only 1 hidden node and two output nodes. The idea is to think of the weights to the hidden node as a 'filter' that can separate two distinct classes. For example, figure 34 shows the weights of the neural net that was trained to separate Class 1 (tree-leaves) from Class 6 (grass-green).

Figure 34 seems to indicate that wider spectral band system could be used to separate of these two classes, However figure 35 shows what should be a simple separation, tree-leaves vs. skin-2. Here there does not seem to be clustering banding and the weight values are noticeably smaller. Also, this network shows a somewhat common behavior for a number of the two class separation networks. Only 1 of the output nodes is projecting the correct output values, +1 for class 1 and -1 for class 2. The other output node is responding with a value that is nearly the same no matter what class the data is from, therefore its mean squared error (MSE) is only 0.99. This network has not trained to completion. Training was stopped because it reached the maximum number of training iterations allowed, which for this particular effort was set to one million.

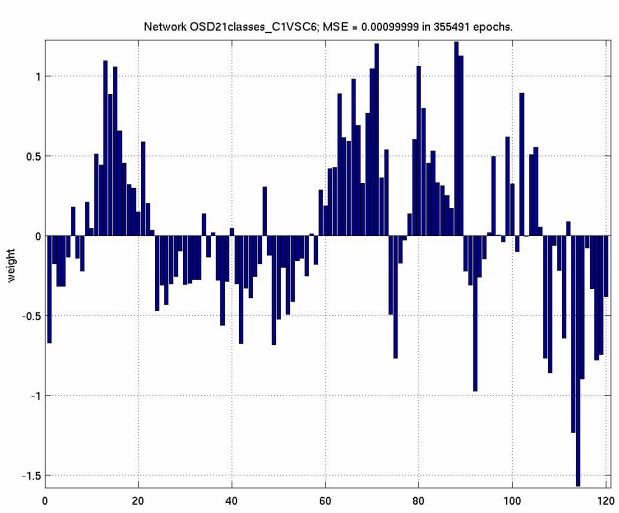


Figure 34. Weights from input to hidden layer. Note clustering of neighboring band weights.

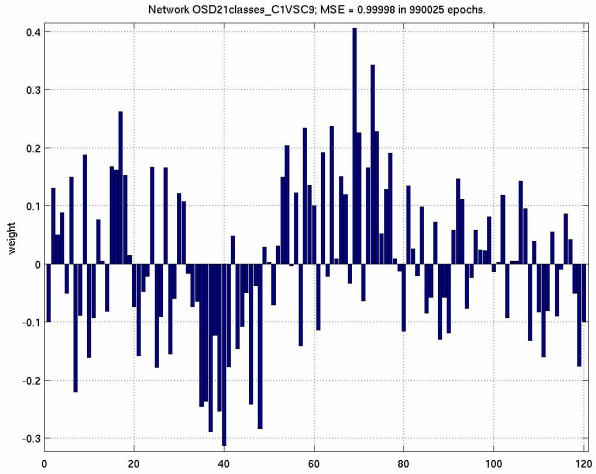


Figure 35. Input to hidden layer weights, for separation of Class 1(tree-leaves) and Class 9 (skin-2)

Looking at the training time may give some indication of how 'easy' the separation between the two classes is. Most, but not all, of the fully trained networks trained very quickly—several thousand to several tens of thousands of training epochs. However, some fully trained networks take much longer, on the order of many hundreds of thousands of training epochs to reach the target MSE of 0.001.

Networks that are only around 0.99 MSE at the conclusion of training seem to be stuck. The question is can anything be done to assist them out of their rut, or with more training will they reach the target MSE of 0.001. There are a few examples where the network took a very long

time, where it seemed stuck and then suddenly it finds it way out and rapidly trains to completion. There are also a number of networks that while not reaching the target MSE, are none the less nearly trained, with MSE on the order of 0.005 or 0.013. It is believed that these networks have found their way toward a complete solution and that with more training they will reach the target MSE of 0.001. We will be exploring that further.

The neural networks show much promise for use in processing hyperspectral imagery, although it is clear that to properly process areas in shadow, some training data for each class should come from shadowed areas. Further, some additional classes are also required. Several images, not shown here, have blue sky in there upper portions. As the NN has not been trained on sky, the closest match is the blue pickup. So in images with sky, processed with the neural networks, the sky is misclassified as blue truck. This should be easily corrected with some samples from the sky.

Work continues with the neural networks. We continue to investigate their performance in both 2 class and multi-class separations. We also continue to investigate how the NN's weight structures might reveal information relevant to possible band selection activities.

One of the issues we have yet to address is the fact that most of the spectral samples we extracted were collected only from relatively few images that have generally the same sun-target-observer angles. So, this data does not address the bi-directional reflectance distribution funtions (BRDF's) of the materials. As most of these materials are not lambertian surfaces, the BRDF is not uniform over the possible geometries. This may also account for some of the false alarms in the different images and may play into the performance in shadows. In fact, this may be the reason that the network used to process figure 27, mis-classifies the truck as the skin-2 class.

5. Semi-parametric Hyperspectral Anomaly Detector

5.1 A Breakthrough in Hyperspectral Anomaly Detection

The image processing challenges posed by this *robotics project* redirected our original applied research plan from merely evaluating conventional anomaly detection techniques using hyperspectral imagery to the unplanned development of unconventional statistical approaches. The original goal of the image processing portion in this project was to evaluate the effectiveness of various anomaly detection algorithms detecting the presence of anomalous structures in natural clutter backgrounds, as a collection of point anomalies. Examples of anomalous structures in this work are stationary personnel, stationary land vehicles, wire fences, and possibly camouflaged objects of interest (targets). The results shown as the proof of principle experiment arguably constitute a breakthrough detecting full pixel targets. Full pixel targets

consist of multiple pixels in the image, contrary to subpixel targets, which are smaller than the spatial resolution of a pixel. (A pixel is the smallest spatial unit in an image.)

A breakthrough that led to the development of our unconventional anomaly detection approach occurred when we discovered that the reason conventional techniques produce high numbers of meaningless detections in digitized scenes (as characterized by an image analyst) is that these techniques do not address explicitly all of the most common spatial/spectral variability occurrences that may be observed locally in the imagery. To better understand the behavior of conventional anomaly detectors, we applied five known techniques on actual HS imagery, including the industry standard, and decomposed the most probable spatial/spectral variability occurrences into three study cases: Case 1, Case 2, and Case 3, where Case 1 represents a comparison between two samples from different distributions (e.g., land vehicle and grass), Case 2 represents a comparison between a two-class sample and a sample of one of the two classes (e.g., a spatial transition between tree shadows and surrounding terrain), and Case 3 represents a comparison between two samples from the same distribution (e.g., grass and grass). We concluded through simulation and inspection of these detectors' performances [5] that the *application* of conventional techniques to local anomaly detection problems is *flawed*. These algorithms are developed to account explicitly for Case 1 and Case 3, but not for one of the most abundant cases—Case 2. Case 2 occurs quite often on digital scenes, they represent major transitions of regions, also strong edges due to the presence of manmade objects in a natural clutter background. This discovery applies to conventional techniques based on parametric or nonparametric approaches. We converted this weakness into strength by proposing an idea for the development of algorithms that can explicitly account for all three study cases: Compare samples indirectly by combining them. This principle is discussed next.

Principle of Indirect Comparison [1]: We proposed to compare samples not as individual entities, but as individual entities and the union among these entities. Let X and Y denote two random samples, denote X a reference sample and let $Z = X \cup Y$, where U denotes the union. Features of the distribution of X can be indirectly compared to features of the distribution of Y by comparing instead features of the distributions of Z and X. We developed and showed that an anomaly detection algorithm based on this principle enjoyed the desirable outcome of preserving what is often characterized by image analysts as meaningful detections (e.g., a manmade object in an open terrain), while significantly reducing the number of meaningless detections (e.g., transition of different regions).

The detector in reference, which uses a semiparametric (SemiP) model, was derived from a solid theoretical background, and—as we will show—can perform remarkably well addressing very difficult anomaly detection tasks, such as, detection of a virtually invisible camouflaged soldier hidden in a natural clutter background—invisible to a casual or critical observer of the scene.

5.2 Semiparametric Anomaly Detector

We describe a semiparametric (SemiP) anomaly detector developed to implement the principle of indirect comparison discussed in the introduction.

Consider the following model, where two random samples x_k ($k = 0,1$) have their components independently, identically distributed (iid) and x_0 is independent of x_1:

$$\begin{aligned} x_1 &= \left(x_{11}, \cdots, x_{1n_1}\right) \ iid \ \sim g_1(x) \\ x_0 &= \left(x_{01}, \cdots, x_{0n_0}\right) \ iid \ \sim g_0(x), \end{aligned} \tag{1}$$

$$\frac{g_1(x)}{g_0(x)} = \exp(\alpha + \beta x) \tag{2}$$

Notice that since g_1 is a density, $\beta = 0$ must imply that $\alpha = 0$, as α merely functions as a normalizing parameter. Notice also that if $\beta = 0$ then x_0 and x_1 belong to the same population (i.e., $g_1 = g_0$). Taking these comments in consideration, an anomaly detector was designed from the following composite hypothesis test:

$$\begin{aligned} H_0: &\quad \beta = 0 \quad (g_1 = g_0) \qquad anomaly \quad absent \\ H_1: &\quad \beta \neq 0 \quad (g_1 \neq g_0) \qquad anomaly \quad present. \end{aligned} \tag{3}$$

The detector tests locally a digitized scene and produces a decision surface, which a binary surface can be generated from by labeling hypothesis H_1 as "1" and H_0 as "0." An isolated object is expected to produce a cluster of "1" values (anomalies) in this binary surface.

The detector is based on the asymptotic behavior of the maximum likelihood estimate (MLE) of $\hat{\beta}$, which can be shown [ARL TR Rosario] to be

$$\sqrt{n}\left(\hat{\beta} - \beta_0\right) \Rightarrow N\left(0, \frac{\rho^{-1}(1+\rho)^2}{Var(t)}\right) \quad \text{as } n \to \infty, \tag{4}$$

where the combined sequence (real valued, not vector valued)

$$t = \left(x_{11}, \cdots, x_{1n_1}, x_{01}, \cdots, x_{0n_0}\right) = \left(t_1, \cdots, t_n\right),$$

$Var(t)$ is the variance of the random sequence t, β_o is the true value of β, $\rho = n_1/n_0$, and $n = n_1 + n_0$. Normalizing the left side of (4) by the maximum likelihood (ML) estimator of $Var(t)$, $\hat{V}(t)$, then setting $\beta_o = 0$, and squaring the final result, one can test H_0 in (3) with the following expression [1]:

$$\chi = n\rho(1+\rho)^{-2}\hat{\beta}^2\hat{V}(t) \xrightarrow[n\to\infty]{} \chi_1^2, \tag{5}$$

which has the chi-square distribution asymptotic behavior with one degree of freedom, χ_1^2 . A decision is based on the value of χ in (5), i.e., high values reject hypothesis H_o, declaring anomalies. A threshold can be selected based on the type I error probability (probability of rejecting H_o, given that H_o is true), which is under the user's control. The expression in (5) constitutes the SemiP anomaly detector.

A practical implementation of the SemiP detector requires maximization of the log likelihood function

$$\log[\zeta(\alpha,\beta,g_0)] = \sum_{j=1}^{n_1}(\alpha+\beta x_{1i}) - \sum_{i=1}^{n}\log[1+\rho\exp(\alpha+\beta t_i)], \tag{6}$$

which is a direct result from the likelihood function

$$\begin{aligned}\zeta(\alpha,\beta,g_0) &= \prod_{i=1}^{n_0} g_0(x_{0i})\prod_{j=1}^{n_1}\exp(\alpha+\beta\ x_{1j})\ g_0(x_{1j}) \\ &= \prod_{i=1}^{n=n_1+n_0} g_0(t_i)\prod_{j=1}^{n_1}\exp(\alpha+\beta\ x_{1j}),\end{aligned} \tag{7}$$

in order to find ML estimators $\hat{\alpha}$ and $\hat{\beta}$ of α and α and β respectively. Both ML estimators are required to find V in (5.5), where it can be shown [2] that

$$\hat{V}(t) = \hat{E}(t^2) - \hat{E}^2(t), \tag{8}$$

where

$$\hat{E}(t^k) = \sum_i t_i^k \hat{g}_0(t_i) \tag{9}$$

and

$$\hat{g}_0(t_i) = \frac{1}{n_0}\frac{1}{1+\rho\exp(\hat{\alpha}+\hat{\beta}t_i)}. \tag{10}$$

5.3 Decision Logic for Multi-class Anomaly Detection

Using the ground work described in Section 2, one can extend that two-sample discussion to allow for a multi-class comparison by adopting, for instance, the following decision logic:

$$Y_{ij} = Min\{\chi_1, \chi_2, \chi_3, \cdots, \chi_L\}$$

where, χ_k is a realization of (5.5), ($k = 1,\ldots,L$; L is the number of object classes), and Y_{ij} is the detector's final result (the minimum of all χ_k) for a given (i,j) location in the image.

For the sake of this discussion, let $L = 5$, where each sample set (consisting of 100 spectra) is assumed to have been manually selected from different objects in a scene and is expected to be compared to samples from the entire imagery. One of the five spectral sample sets is arbitrarily chosen as the *reference object*. The spectral angle between each spectral vector (reference object) and the reference object's spectral average is computed to form one of the *reference feature sets*. The other four reference feature sets are formed by computing the spectral angles between the reference spectral vectors and the spectral averages of the other four sampled objects in the scene. From the *5* resulting reference feature sets, each one shall be tested against a *test feature set,* which shall be formed by computing the spectral angles between each one of the reference spectral sample and the spectral average from a given location at the image. At a given location (i,j), χ_k $(k = 1,\ldots,5)$ will be computed producing a result Y_{ij}. This procedure will be applied to the entire scene. One should notice that a local anomaly is expected to produce relative high values of Y_{ij}, revealing ultimately an anomalous structure with respect to the five most dominant classes in that scene.

5.4 Proof of Principle Results

The data used in this work was collected with a novel visible to near-IR spectral imager (SOC-700) from Surface Optics Corporation, San Diego, CA. The system is a relatively small, portable hyperspectral imager, which collects a hyperspectral cube consisting of 640 x 640 pixels x 120 spectral bands and has a spectral range covering 0.38 to 0.92 μm. The sensor is commercially available off the shelf [3].

The data were collected during the month of June 2004 in Fort Hunter-Liggett, CA, to support a research effort between ARL and Army RDECOM-ARDEC. Three scenes from that data collection were used for this study. The first row in figure 36 shows the photos of those scenes, which were taken using a standard digital photo camera, and the second row depicts those scenes as the average of 120 bands, which were collected using the SOC-700 HS camera.

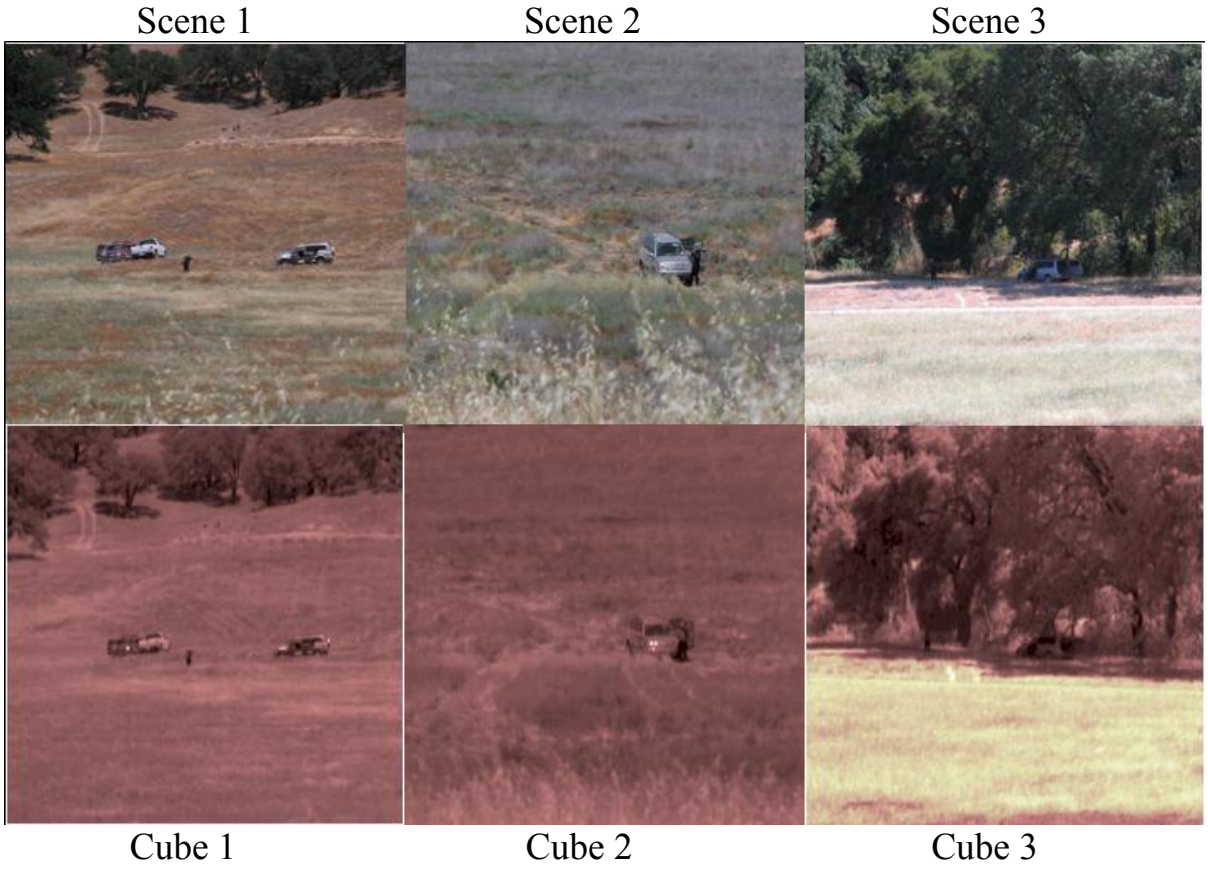


Figure 36. Scene photos at the top and their corresponding SOC-700 hyperspectral cubes (band averages).

From actual ground truth, it is known that Scene 1 contains three motor vehicles and a standing person in the center of that scene (i.e., two pick-up trucks to the left in proximity to each other, a man slightly forward from the vehicles in the center, and a sport utility to the right). The background in Scene 1 is dominated by the following two most abundant object classes: Californian valley-type trees and terrain. (We arbitrarily selected Scene 1 to draw two sets of spectral samples—one consisting of tree samples and the other of terrain samples—and the goal is to use these spectral sets as references to test Cube 1, Cube 2, and Cube 3; the locations of these sample sets are shown as yellow boxes in Cube 1, fig. 37.) Scene 2 consists of the same sport utility vehicle and the same person standing in proximity to it, they are located in the same valley, but at a different area from the one in Scene 1. Scene 3 depicts a significantly more complex scenario, where, from actual ground truth, it is known that the same man and the same sport utility vehicle can be found in the shadows of a large cluster of trees. Portions of the shadowed vehicle are observable near the center in Scene 3, and the shadowed man is reportedly halfway between the vehicle and the left side of the image. The three scenes were independently displayed using the same *color map*, which means that the pixel intensities shown in each individual surface is only relative to the corresponding values in that surface. (Pixel values representing the same object class may be displayed with different intensities in another surface,

depending on other objects in the scene.) This fact explains the brightness difference between the terrains displayed in Scene 2 and Scene 3, given that the meteorological conditions and terrains were about the same for all three scenes. The strong reflections from certain parts of the vehicles captured by the sensor in Scene 1 and 2 are not as dominant in Scene 3 because the vehicle there is in the shadow; hence, the terrain in Scene 3 appears to be a strong reflector.

In essence, we would like to capture in this study the overall behavior of an anomaly detector in twofold: (i) seeking for global anomalies in a natural clutter background, given that only a few spectral samples from the most abundant object classes in the background (in this case, trees and terrain) are drawn from the same HS data and presented, as references, to the detectors and (ii) seeking for global anomalies, given that the reference samples are not drawn from the same HS data. We would be able to determine in (i) the effectiveness of these detectors within the same data of a particular area in the valley and in (ii) their effectiveness and robustness using data from different areas in the valley.

In figure 37, we present the raw output surfaces produced by two detectors testing the scenes described above: SemiP anomaly detector and adaptive RX anomaly detector [4]. The output surfaces of the RX and SemiP detectors are shown in columns 2 and 3, respectively, for the corresponding HS cubes in column 1. We used a suitable *color map* to emphasize anomalies with respect to the reference samples by their false-color (intensity) levels, i.e., white is equivalent to the strongest anomalies, yellow to strong anomalies, red to intermediate anomalies, brown to weak anomalies, black to weakest anomalies. The false colors change gradually and are relative only to those results within the same surface, for instance, a yellow pixel in one surface does not mean necessarily that its value is equivalent to another yellow pixel in another surface.

The local results shown in the first RX surface (row 1, column 2) are consistent with the study cases discussed in the background. A detector based on conventional methods performs well suppressing objects in the scene having low variability and belonging to the same class of a reference set (*Case 3*)—the trees were suppressed. Likewise, it performs well accentuating objects that are significantly different from the reference set (*Case 1*)—for instance, some parts of the vehicle at the right hand side (row 1, column 2) were highly accentuated. (One can actually observe white pixels within the boundaries of those vehicles by zooming close enough on both RX surfaces (rows 1 and 2, column 2), which indicates that those portions are significantly different from the reference sets. Unfortunately, as it was observed in the top-view problem, local areas characterized by class mixtures (transition of regions) may be also accentuated by these detectors, obscuring therefore the presence of meaningful objects in that scene. In fact, for the HS cubes presented in figure 37, the RX detector seems to perform more as an edge detector than as an anomalous object detector.

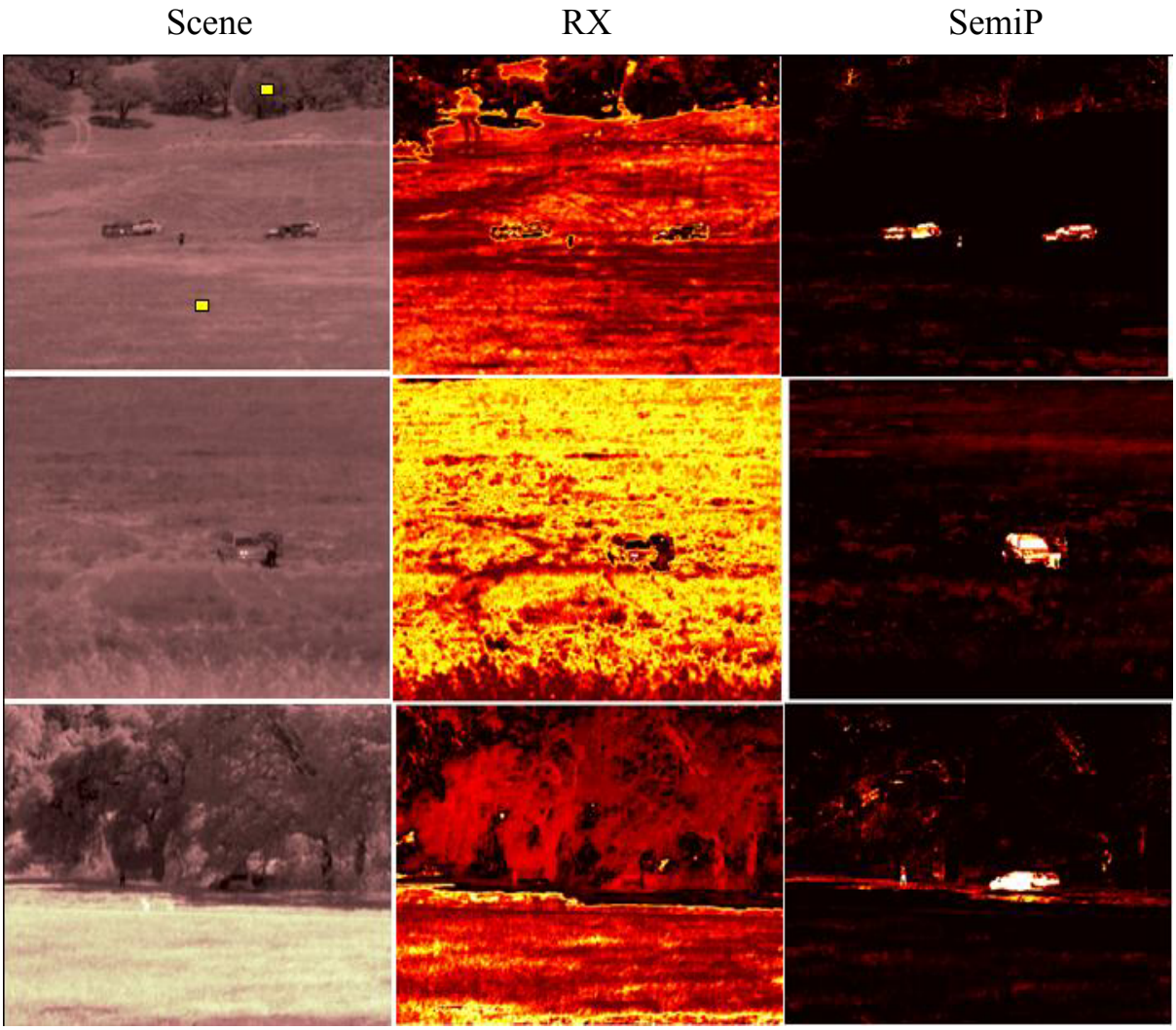


Figure 37. Scene anomaly detection using two reference sets of spectral samples (their locations are shown as yellow boxes in the top scene) from California tree leaves and valley terrain. The unconventional SemiP anomaly detector was developed for the robotics project, and the conventional RX anomaly detector is the standard technique for anomaly detection. The RX and SemiP output surfaces are displayed using the same pseudo color map, where *white* depicts the strongest sign of anomalies, *yellow* strong, *red* intermediate, and *black* lowest.

The SemiP detector, on the other hand, was able to suppress virtually all the background of Cube 1, and to accentuate large portions of the vehicles and of the standing man. In a qualitative sense, test samples consisting of, say, a mixture of shadow and terrain were likely suppressed due to the indirect comparison between the mixture itself and the union between that mixture and a component of that mixture, in this case, terrain.

Let's examine next the impact of shadowed objects when compared to a reference consisting of the non-shadowed version of the same object. The reason our indirect comparison based detectors work so well suppressing shadowed patches in the ground may be explained by the following: Regions characterized by tree shadows, for instance, may be interpreted as *partially*

obscured terrain because tree leaves do partially obscure the incident solar light; however, since significant spectral radiances are still reflected from the partially shadowed terrain, such a region will be suppressed when compared to the union of itself and the reference set of open terrain.

Consider now the results shown for Cube 2 in figure 37 (row 2, column 2 and 3). The RX surface shown in rows 2, column 2, suggests that the RX detector may be susceptible to subtle spectral differences of the same terrain when observed by the same HS sensor in a different area. Recall that Scenes 2 and 3 were tested using the same reference sets drawn from Scene 1. The surface shown in row 2, column 3, suggests that the SemiP detector is significantly more robust to spectral differences of the same terrain. The concern of such robustness was addressed as one of our examination goals cited in (ii).

Shifting our attention now to the results corresponding to Cube 3 in figure 37 (row 3, column 2 and 3), the interpretation of a shadowed object as a partially obscured object is especially relevant to the interpretation of output results for Scene 3. The output surface shown in figure 37, row 3, column 2, emphasizes the fact that the RX anomaly detector performs as expected: it detects local anomalies in the scene. And as it was mentioned earlier, these local anomalies are not guaranteed to be meaningful to an image analyst in the context of our problem. For instance, using HS Cube 3, notice that some of the tracks made by the shadowed vehicle, and the transition between the shadowed and the non-shadowed terrain were the most anomalous regions in the scene, as *seen* by the RX detector. Fortunately, with the indirect comparison approach that is inherent in the SemiP detector, these same regions were virtually suppressed, while the more meaningful anomalous structures (vehicle and human pants) were accentuated; see the corresponding SemiP surface in figure 37 (row 3, column 3).

5.5 Extension to Unsupervised Learning Self Classification

It is well known that an effective anomaly detection technique may be adapted to function as an unsupervised learning classifier. In this section, we adapt our indirect comparison, anomaly detection approach to function as a *self classifier* and present a proof-of-principle experiment. Figure 38 depicts the concept. The notion of self classification, in the context of our discussion, simply means that a given algorithm suite consisting of two stages can be used to detect meaningful objects (stage 1) as a collection of point anomalies in respect to some reference set (available *a priori*), and, upon applying a clustering algorithm to separate these detections as mutually exclusive clusters, the anomaly detection engine would be reused to function as a classifier by reintroducing to the anomaly detector samples from the clustered detections, as references, and—relying on the anomaly detector's ability to discriminate—to determine the classification among the detected objects. For instance, suppose that samples from two objects are detected and then clustered into two groups, Class 1 and Class 2. It would be of further interest to know rather Class 1 and Class 2 are the same or different classes. If these classes are the same, one could use the same color code to indicate this fact; else, the two classes would be displayed with different colors. Using this procedure, it would be appealing to have, for instance,

land vehicles that are detected in the same HS cube being able to retain the same color code, a color code that would be different from the one obtained by a different object class (e.g., human beings) also present in the scene. It is worth mentioning that self-classification does not provide information on the actual classes of the objects, albeit it separates objects by class membership using the discriminatory power of the anomaly detector (given of course that these objects are in fact separable).

The output results shown in figure 38 depict this notion of self-classification using the SemiP anomaly detector. The red surface in figure 38 (lower right hand corner) represents the SemiP detector's output surface for Cube 1. That surface clearly shows strong anomaly peaks due to the presence of the vehicles and to the standing person in the open field. Based on the proximity of the two vehicles in the left of the person (reader's perspective), they seem to form a single anomalous object, which is labeled at this preliminary stage as class 1. The person was labeled at this stage as belonging to another class, 2. And the vehicle at the right of the person was labeled as belonging to a third class (labeled as 3). The surface immediately above the red surface is exactly the same surface, but displayed as a 2D surface using a different color map. The circles around the anomalous structures were put artificially in this 2D surface to emphasize the fact that automatic post processing can be applied to exploit the detection markers and to spatially bound each unknown individual object.

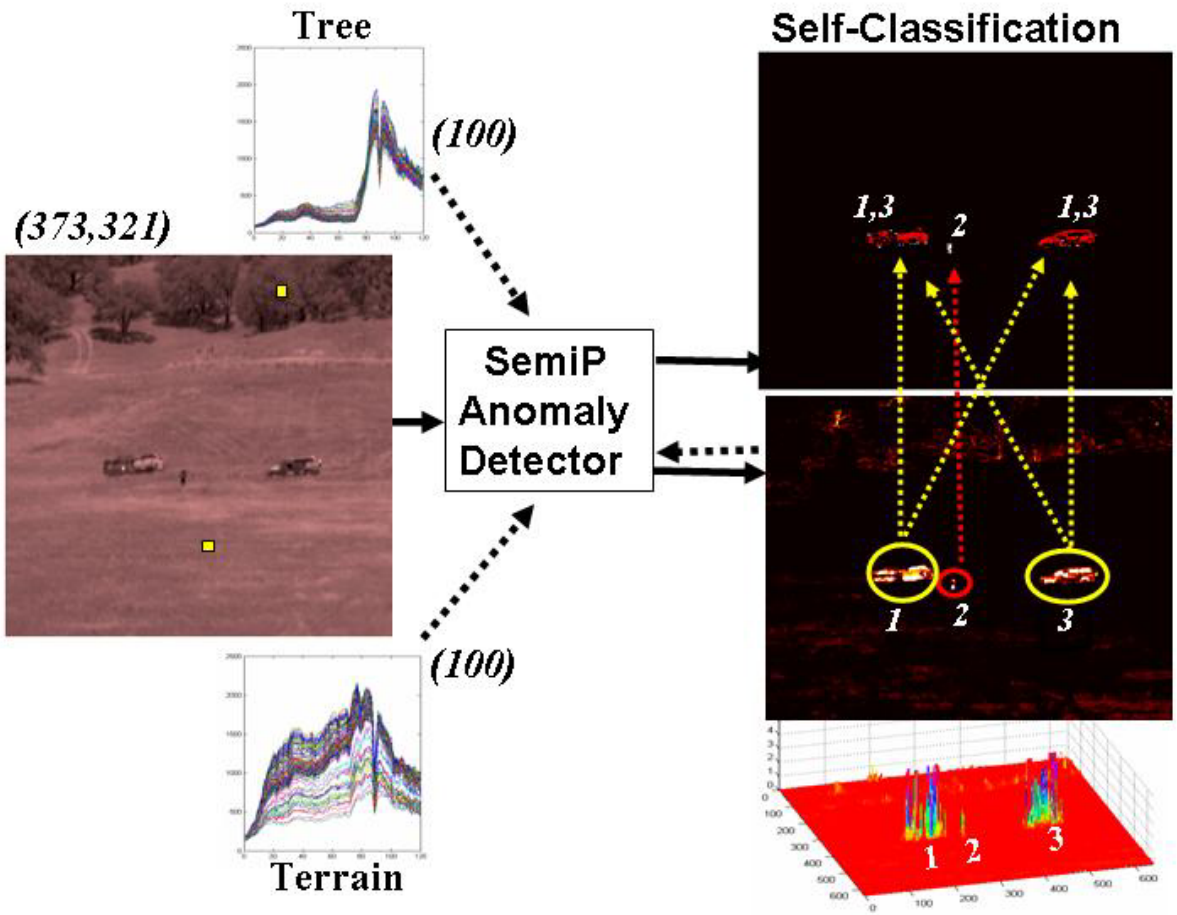


Figure 38. Proof of principle experiment to illustrate a concept of object self-classification using the SemiP anomaly detector twice in the loop.

We used standard morphological filters (i.e., logical combinations of *dilation* and *erosion* to function as an opening operation to reduce noise, and as a closing operation to fill holes within the same object) to produce a silhouette for each class using the most dominant peaks as detection markers.

Spectral samples from within each silhouette (class 1, 2, or 3) were used as a reference set—one at a time—through the same anomaly detector to decide whether the other two classes belonged to the reference class. There could only have five outcomes: (i) the three classes are about the same, (ii) the three classes are significantly different, (iii) classes 1 and 2 are about the same but different from class 3, (iv) classes 1 and 3 are about the same but different from class 2, or (v) classes 2 and 3 are about the same but different from 1.

The blue surface shown in figure 38 shows the final result from the self classification procedure just described, where the blue region represents the suppressed clutter background after stage 1. The SemiP detector produced the outcome (iv), i.e., the vehicles fell into the same class

(depicted by yellow) and an overwhelming portion of the standing person fell into a different class (depicted by red). In summary, using initially two sample sets as references drawn from the scene shown in figure 38, the SemiP anomaly detector was able to find three spatially independent objects as scene anomalies and could conclude that two of them (the three vehicles) belonged to the same class, and the remainder one (the standing person) most likely belonged to a class of its own.

Figure 39 further clarifies the output of the SemiP self classifier. For instance, the surface at the upper left hand side in figure 39 is the output result produced after stage 1 and the post processing procedure that spatially clustered the mutually exclusive objects, as *seen* by the anomaly detector.

One can interpret the joint functions of anomaly detection and the follow-on post image processing as the extraction of meaningful objects from the scene, or as a meaningful focus of attention. This interpretation is emphasized by the 3D surface shown at the upper right hand side in figure 39. The spectral samples shown at the lower right and lower left hand sides are samples of the corresponding objects, person and right side vehicle, and were drawn from the scene using the detection masks (produced by post image processing) shown in figure 39 as white solid shapes. The vehicles at the left side of the person have similar spectral responses. The final output surface produced by reintroducing the new reference sets of spectral samples from the vehicles at the left, from the vehicle at the right, and from the standing person to the SemiP detector is shown at the lower center in figure 39.

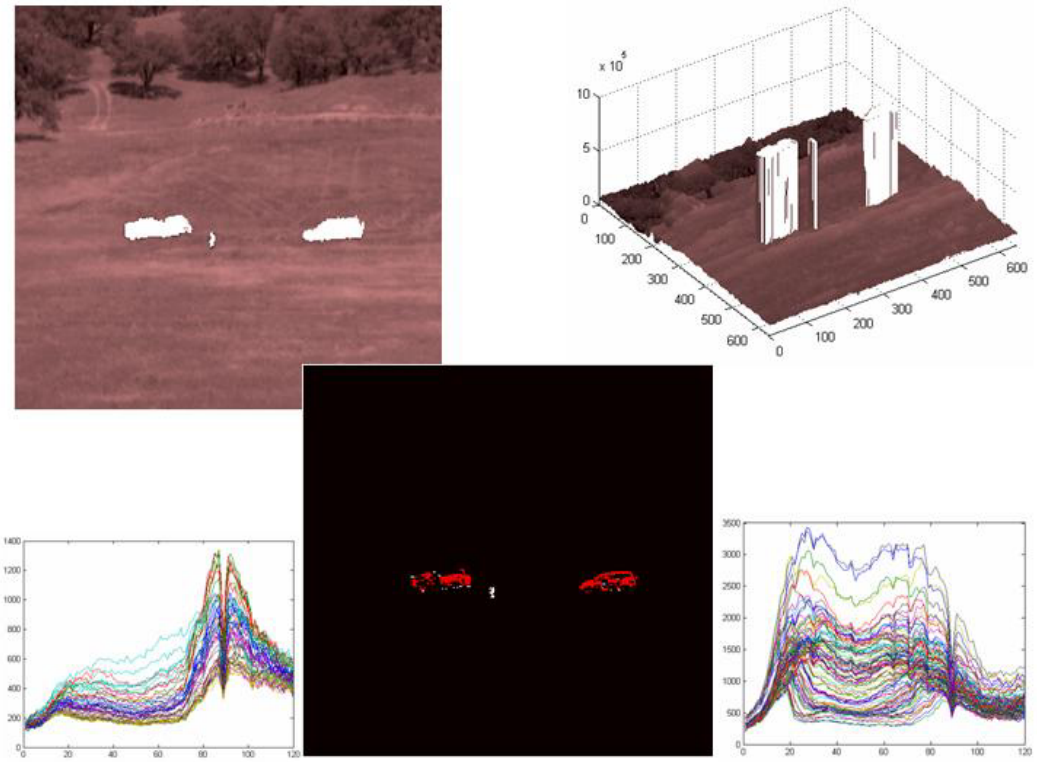


Figure 39. Additional results from a proof of principle experiment to illustrate a concept of self classification using the SemiP anomaly detector twice in the loop.

6. Polarimetric and Kernel-Based Nonlinear Anomaly (Target) Detection

In this section, hyperspectral anomaly (target) detection techniques developed at the EO/IR Image Processing Branch are applied to hyperspectral images generated by SECOTS and DBHSI described in the previous sections, operating at the visible and infrared (IR) bands, respectively. Both the anomaly detection and target classification techniques have been used to detect potential obstacles to UGV navigation such as a wire (e.g. a chain-link fence). The developed anomaly detection techniques include a polarimetric anomaly detection technique and a nonlinear version of the well-known RX detector that detect anomalies (potential targets) without referring to reference spectral signatures. A nonlinear version of the spectral matched filter detector is also developed to detect materials of interest.

6.1 Polarimetric Anomaly Detector

Since the AOTF hyperspectral sensor (SECOTS) has polarization capability, polarimetric hyperspectral images (horizontal and vertical polarization) are produced by SECOTS to detect specular objects located within the field of view (FOV) which are potentially hazardous to UGVs. Two polarimetric hyperspectral images – horizontal and vertical – are used. A polarimetric detection algorithm, based on statistical distance measures between the two polarimetric images, has been developed to detect a barbed wire, a chain-link type gate or a body of water.

In order to exploit the polarimetric difference, a certain statistical distance is measured between the local areas of the two polarimetric hyperspectral images. For each test pixel location in the two hyperspectral images, two small rectangular windows are used simultaneously which are placed at the same pixel locations, as shown in figure 40. The well- known Bhattacharyya distance is then measured between the spectral vectors obtained from the two local windows. The Bhattacharyya distance is given by

$$\mathbf{BD} = \frac{1}{8}(\mathbf{M}_2 - \mathbf{M}_1)^{\mathbf{T}}\left(\frac{\mathbf{C}_1 + \mathbf{C}_2}{2}\right)^{-1}(\mathbf{M}_2 - \mathbf{M}_1) + \frac{1}{2}\ln\frac{|\mathbf{C}_1 + \mathbf{C}_2|}{\sqrt{|\mathbf{C}_1||\mathbf{C}_2|}},$$

where $\mathbf{M}_1$ and $\mathbf{C}_1$ are the mean vector and covariance matrix, respectively, of the local spectral vectors within the local window for the horizontally polarized hyperspectral image. Similarly, $\mathbf{M}_2$ and $\mathbf{C}_2$ are the mean vector and covariance matrix, respectively, for the vertically polarized hyperspectral image. If the distance is larger than a predefined threshold the test pixel is considered an anomaly.

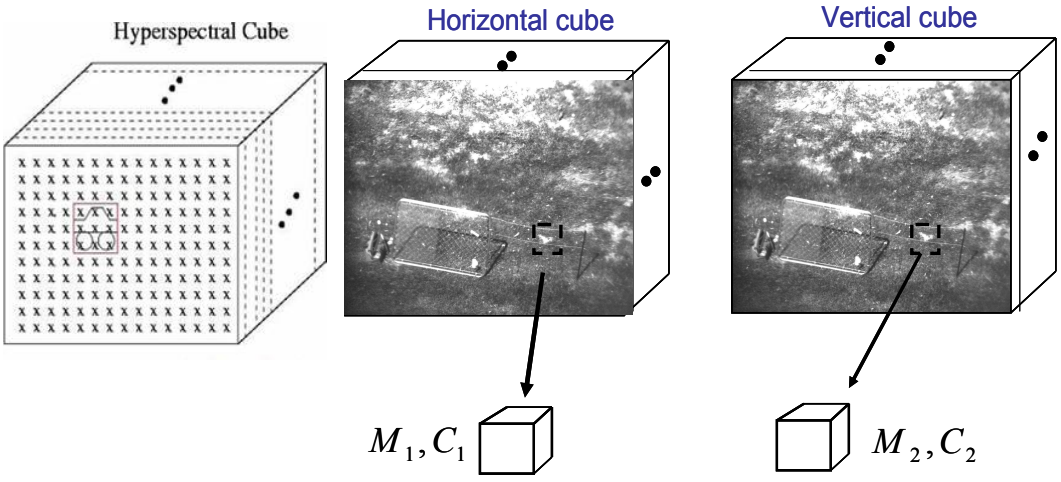


Figure 40. Polarimetric anomaly detection.

6.2 Kernel-based Nonlinear Detectors

In this section, we present two types of nonlinear kernel-based target detection methods: a nonlinear version of the well-known RX anomaly detection method referred to as the kernel RX-

algorithm [1] and the kernel spectral matched filter detector. Both kernel-based algorithms are obtained by extending the corresponding conventional algorithms into a feature space associated with the original input space via a certain nonlinear mapping function. The kernel-based detectors exploit nonlinear correlations between the spectral bands that are ignored by the conventional detectors. Nonlinear realization is mainly pursued to reduce data complexity in a high dimensional feature space and consequently provides simpler decision rules for data discriminations. In this section, we show how each of the kernel-based algorithms is derived by using the *kernel trick* that implicitly computes the dot products in the feature space using kernels.

6.2.1 Kernel Methods and the Kernel Tricks

Suppose the input hyperspectral data is represented by the data space $\left(\mathbf{X} \subseteq \mathbf{R}^{\mathbf{l}}\right)$ and **F** is a feature space associated with **X** by a nonlinear mapping function $\mathbf{\Phi}$

$$\mathbf{\Phi} : \mathbf{X} \to \mathbf{F}, \mathbf{x} \mapsto \mathbf{\Phi}(\mathbf{x}) \quad , \tag{11}$$

where **x** is an input vector in **X** which is mapped into a potentially much higher (could be infinite) dimensional feature space. Due to the high dimensionality of the feature space **F**, it is computationally not feasible to implement any algorithm directly in feature space. However, kernel-based learning algorithms use an effective kernel trick given by equation (12) to implement dot products in feature space by employing kernel functions [2]. The idea in kernel-based techniques is to obtain a nonlinear version of an algorithm defined in the input space by implicitly redefining it in the feature space and then converting it in terms of dot products. The kernel trick is then used to implicitly compute the dot products in F without mapping the input vectors into F; therefore, in the kernel methods, the mapping $\mathbf{\Phi}$ does not need to be identified.

The kernel representation for the dot products in F is expressed as

$$\mathbf{k}(\mathbf{x_i}, \mathbf{x_j}) = \mathbf{\Phi}(\mathbf{x_i}) \cdot \mathbf{\Phi}(\mathbf{x_j}) \quad , \tag{12}$$

where **k** is a kernel function in terms of the original data. There are a large number of Mercer kernels that have the kernel trick property. See [2] for detailed information about the properties of different kernels and kernel-based learning. Our choice of kernel in this report is the Gaussian RBF kernel and the associated nonlinear function $\mathbf{\Phi}$ with this kernel generates a feature space of infinite dimensionality.

6.2.2 Kernel-RX Anomaly Detector

6.2.2.1 Introduction to RX-Algorithm

Reed and Yu in [3] developed a generalized likelihood ratio test (GLRT), so called the RX anomaly detection, for multidimensional image data assuming that the spectrum of the received signal (spectral pixel) and the covariance of the background clutter are unknown. Let each input

spectral signal be denoted by a vector **x** consisting of J spectral bands. Define $\mathbf{X}_b$ to be a $J \times M$ matrix of the M reference background clutter pixels. Each observation spectral pixel is represented as a column in the sample matrix $\mathbf{X}_b$.

The two competing hypotheses that the RX-algorithm must distinguish are given by

$$\begin{aligned} &\mathbf{H}_0 : \mathbf{x} = \mathbf{n}, && \text{Target absent} \\ &\mathbf{H}_1 : \mathbf{x} = a\mathbf{s} + \mathbf{n}, && \text{Target present} \end{aligned} \tag{13}$$

where a=0 under $\mathbf{H}_0$ and a=1 under $\mathbf{H}_1$, respectively. $\mathbf{n}$ is a vector that represents the background clutter noise process, and $\mathbf{s}$ is the spectral signature of the signal (target). The target signature $\mathbf{s}$ and background covariance $\mathbf{C}_b$ are assumed to be unknown. The model assumes that the data arises from two normal PDFs with the same covariance matrix but different means. Under $\mathbf{H}_0$ the data (background clutter) is modeled as $N(0, \mathbf{C}_b)$ and under $\mathbf{H}_1$ it is modeled as $N(\mathbf{s}, \mathbf{C}_b)$. The background covariance $\mathbf{C}_b$ is estimated from the reference background clutter data. The estimated background covariance $\hat{\mathbf{C}}_b$ is given by

$$\hat{\mathbf{C}}_b = \frac{1}{M}(\mathbf{x}(i) - \hat{\boldsymbol{\mu}}_b)(\mathbf{x}(i) - \hat{\boldsymbol{\mu}}_b)^T,$$

Where $\hat{\boldsymbol{\mu}}_b$ is the estimated background clutter sample mean given by $\hat{\boldsymbol{\mu}}_b = \frac{1}{M}\sum_{i=1}^{M}\mathbf{x}(i)$. Assuming a single pixel target **r** as the observation test vector, the expression for the RX-algorithm is given by

$$RX(\mathbf{r}) = (\mathbf{r} - \hat{\boldsymbol{\mu}}_b)^T \hat{\mathbf{C}}_b^{-1} (\mathbf{r} - \hat{\boldsymbol{\mu}}_b).$$

6.2.2.1 Kernel-RX Algorithm

In this section, we remodel the RX-algorithm in the feature space by assuming the input data has already been mapped into a high dimensional feature space. The two hypotheses in the nonlinear domain are now

$$\begin{aligned} &\mathbf{H}_{0_\Phi} : \boldsymbol{\Phi}(\mathbf{x}) = \boldsymbol{\Phi}(\mathbf{n}), && \text{Target absent} \\ &H_{1_\Phi} : \boldsymbol{\Phi}(\mathbf{x}) = \mathbf{a}_\Phi \boldsymbol{\Phi}(\mathbf{s}) + \boldsymbol{\Phi}(\mathbf{n}), && \text{Target present} \end{aligned} \tag{14}$$

The corresponding RX-algorithm in the feature space is

$$\mathbf{RX}(\boldsymbol{\Phi}(\mathbf{r})) = (\boldsymbol{\Phi}(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi})^T \hat{\mathbf{C}}_{b_\Phi}^{-1} (\boldsymbol{\Phi}(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{b_\Phi}) \tag{15}$$

where $\hat{\mathbf{C}}_{b_\Phi}^{-1}$ and $\hat{\boldsymbol{\mu}}_{b_\Phi}$ are the estimated covariance and background clutter sample mean in the feature space, respectively, given by

$$\hat{\mathbf{C}}_{\mathbf{b}_\Phi} = \frac{1}{\mathrm{M}}\sum(\mathbf{\Phi}(\mathbf{x}(\mathrm{i}) - \hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi})(\mathbf{\Phi}(\mathbf{x}(\mathrm{i}) - \hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi})^{\mathbf{T}} \quad \text{and} \quad \hat{\mu}_{b_\Phi} = \frac{1}{M}\sum_{i=1}^{M}\Phi(x(i)).$$

The nonlinear RX-algorithm given by equation (15) is now in the feature space which cannot be implemented explicitly due to the non-linear mapping $\mathbf{\Phi}$ which produces a data space of high dimensionality. In order to avoid implementing Equation (15) directly we need to *kernelize* Equation (15) by using the kernel trick introduced in Section 6.1.

The estimated background covariance matrix can be represented by its eigenvector decomposition or spectral decomposition as given by

$$\hat{\mathbf{C}}_{\mathbf{b}_\Phi} = \mathbf{V}_\Phi \mathbf{\Lambda}_{\mathrm{b}} \mathbf{V}_\Phi^{\mathrm{T}},$$

where $\mathbf{\Lambda}_{\mathrm{b}}$ is a diagonal matrix consisting of the eigenvalues and $\mathbf{V}_\Phi$ is a matrix whose columns are the eigenvectors of $\hat{\mathbf{C}}_{\mathbf{b}_\Phi}$ in the feature space. The eigenvector matrix $\mathbf{V}_\Phi$ is given by $\mathbf{V}_\Phi = [\mathbf{v}_\Phi^1, \mathbf{v}_\Phi^2, \cdots]$, where $\mathbf{v}_\Phi^{\mathbf{j}}$ is the *j*th eigenvector with non-zero eigenvalue.

The pseudo-inverse of the estimated background covariance matrix can also be written as

$$\hat{\mathbf{C}}_{\mathbf{b}_\Phi}^{\#} = \mathbf{V}_\Phi \mathbf{\Lambda}_{\mathbf{b}}^{-1} \mathbf{V}_\Phi^{\mathrm{T}}. \tag{16}$$

Each eigenvector $\mathbf{v}_\Phi^{\mathbf{j}}$ in the feature space can be expressed as a linear combination of the centered input vectors $\mathbf{\Phi}_{\mathbf{c}}(\mathbf{x}(\mathbf{i})) = \mathbf{\Phi}(\mathbf{x}(\mathbf{i})) - \hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi}$ in the feature space as shown by $v_\Phi^j = \sum_{i=1}^{M}\beta_i^j \Phi_c(x(i)) = X_{b_\Phi}\beta^j$ where $\mathbf{X}_{\mathbf{b}_\Phi} = [\mathbf{\Phi}_{\mathbf{c}}(\mathbf{x}(1)), \mathbf{\Phi}_{\mathbf{c}}(\mathbf{x}(2)), \cdots, \mathbf{\Phi}_{\mathbf{c}}(\mathbf{x}(\mathbf{M}))]$ and for all the eigenvectors and $\boldsymbol{\beta}^{\mathbf{j}}, \mathbf{j} = 1,2,\cdots,\mathbf{M}$, are eigenvectors of the kernel matrix (Gram matrix) $\hat{\mathbf{K}}(\mathbf{X}_{\mathbf{b}}, \mathbf{X}_{\mathbf{b}})$ normalized by the square root of their corresponding eigenvalues. For all the eigenvectors with nonzero eigenvalues

$$V_\Phi = X_{b_\Phi}\mathrm{B}, \tag{17}$$

where $\mathbf{B} = (\boldsymbol{\beta}^1, \boldsymbol{\beta}^2, \cdots, \boldsymbol{\beta}^{\mathbf{M}})^{\mathbf{T}}$.

Substituting equation (17) into (16) yields

$$\hat{C}_{b_\Phi}^{\#} = X_{b_\Phi}\mathrm{B}\Lambda_b^{-1}\mathrm{B}^T X_{b_\Phi}^T. \tag{18}$$

Inserting equation (18) into equation (15) the nonlinear RX-algorithm can be rewritten as

$$\mathbf{RX}(\mathbf{\Phi}(\mathbf{r})) = (\mathbf{\Phi}(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi})^{\mathbf{T}} \mathbf{X}_{\mathbf{b}_\Phi} \mathbf{B} \mathbf{X}_{\mathbf{b}_\Phi}^{\mathbf{T}} (\mathbf{\Phi}(\mathbf{r}) - \hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi}) \tag{19}$$

The dot product terms $\mathbf{\Phi}(\mathbf{r})^{\mathbf{T}}\mathbf{X}_{\mathbf{b}_\Phi}$ in the feature space can be represented in terms of the kernel function:

$$\Phi(\mathbf{r})^{\mathbf{T}}\mathbf{X}_{\mathbf{b}_\Phi} = \mathbf{k}(\mathbf{X}_\mathbf{b},\mathbf{r})^{\mathbf{T}} - \frac{1}{\mathbf{M}}\sum_{\mathbf{i}=1}^{\mathbf{M}}\mathbf{k}(\mathbf{x}(\mathbf{i}),\mathbf{r}) \equiv \mathbf{K}_\mathbf{r}^{\mathbf{T}}, \tag{20}$$

where $\mathbf{k}(\mathbf{X}_\mathbf{b},\mathbf{r})^{\mathbf{T}}$ represents a vector whose entries are the kernels $\mathbf{k}(\mathbf{x}(\mathbf{i}),\mathbf{r}), \mathbf{i} = 1,\cdots,\mathbf{M}$, and $\frac{1}{\mathbf{M}}\sum_{\mathbf{i}=1}^{\mathbf{M}}\mathbf{k}(\mathbf{x}(\mathbf{i}),\mathbf{r})$ represents the scalar mean of $\mathbf{k}(\mathbf{X}_\mathbf{b},\mathbf{r})^{\mathbf{T}}$.

Similarly,

$$\hat{\boldsymbol{\mu}}_{\mathbf{b}_\Phi}^{\mathbf{T}}\mathbf{X}_{\mathbf{b}_\Phi} = \frac{1}{\mathbf{M}}\sum\mathbf{k}(\mathbf{x}(\mathbf{i}),\mathbf{X}_\mathbf{b}) - \frac{1}{\mathbf{M}^2}\sum_{\mathbf{i}=1}^{\mathbf{M}}\sum_{\mathbf{j}=1}^{\mathbf{M}}\mathbf{k}(\mathbf{x}(\mathbf{i}),\mathbf{x}(\mathbf{j})) \equiv \mathbf{K}_{\hat{\boldsymbol{\mu}}_\mathbf{b}}^{\mathbf{T}} \tag{21}$$

Also using the properties of the Kernel PCA [4], we have the relationship

$$\hat{\mathbf{K}}_\mathbf{b}^{-1} = \frac{1}{\mathbf{M}}\mathbf{B}\boldsymbol{\Lambda}_\mathbf{b}^{-1}\mathbf{B}^{\mathbf{T}}, \tag{22}$$

where we denote the estimated centered Gram matrix $\hat{\mathbf{K}}_\mathbf{b} = \hat{\mathbf{K}}(\mathbf{X}_\mathbf{b},\mathbf{X}_\mathbf{b}) = (\hat{\mathbf{K}})_{\mathbf{ij}}$

the M$\times$ M kernel matrix whose entries $\mathbf{k}(\mathbf{x}_\mathbf{i},\mathbf{x}_\mathbf{j})$ are the dot products $<\Phi_\mathbf{c}(\mathbf{x}_\mathbf{i})\cdot\Phi_\mathbf{c}(\mathbf{x}_\mathbf{j})>$ and M is the total number of background clutter samples which can be ignored. Substituting equations (20), (21), and (22) (without $\frac{1}{\mathbf{M}}$) into equation (19) the kernelized version of the RX-algorithm is given by

$$\mathbf{RX}_\mathbf{K}(\mathbf{r}) = (\mathbf{K}^{\mathbf{T}} - \mathbf{K}_{\hat{\boldsymbol{\mu}}_\mathbf{b}}^{\mathbf{T}})^{\mathbf{T}}\hat{\mathbf{K}}_\mathbf{b}^{-1}(\mathbf{K}_\mathbf{r}^{\mathbf{T}} - \mathbf{K}_{\hat{\boldsymbol{\mu}}_\mathbf{b}}^{\mathbf{T}}) \tag{23}$$

which can now be implemented with no knowledge of the mapping function Φ. The only requirement is a good choice for the kernel function **k**.

Note that $\hat{\mathbf{K}}_\mathbf{b}$ is the centered Gram matrix, as shown in [2].

The centered $\hat{\mathbf{K}}_\mathbf{b}$ is given by

$$\hat{\mathbf{K}}_\mathbf{b} = (\mathbf{K}_\mathbf{b} - 1_\mathbf{N}\mathbf{K}_\mathbf{b} - \mathbf{K}_\mathbf{b}1_\mathbf{N} - 1_\mathbf{N}\mathbf{K}_\mathbf{b}1_\mathbf{N}),$$

where $\mathbf{K}_\mathbf{b}$ is the Gram matrix before centering and $(1_\mathbf{N})_{\mathbf{ij}} = \frac{1}{\mathbf{N}}$ is an N $\times$ N matrix.

6.2.3 Kernel Spectral Matched Filter

6.2.3.1 Linear Spectral Matched Filter

In this section, we introduce the concept of linear spectral matched filter (SMF). The constrained least squares approach is used to derive the linear SMF. Suppose the input spectral signal **x**

consists of p spectral bands. We can model each spectral observation as a linear combination of the target spectral signature and noise $\mathbf{x} = a\mathbf{s} + \mathbf{n}$, where a is an attenuation constant (target abundance measure). When a=0 no target is present and when a > 0 target is present, vector **s** contains the spectral signature of the target and vector **n** contains the added background clutter noise.

Let us define **X** to be a p $\times$ N matrix of the N mean-removed background reference pixels (centered) obtained from the input image. Let each centered observation spectral pixel to be represented as a column in the sample matrix **X** $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n]$. We can design a linear matched filter **w** such that the desired target signal **s** is passed through while the average filter output energy is minimized. The solution to this minimization problem was shown in [5] and was called Constrained Energy Minimization (CEM) filter given by

$$\mathbf{w} = \frac{\hat{\mathbf{C}}^{-1}\mathbf{s}}{\mathbf{s}^{\mathbf{T}}\hat{\mathbf{C}}^{-1}\mathbf{s}}$$

where the estimated covariance matrix $\hat{\mathbf{C}}$ for the mean-removed (centered) reference data is given by $\hat{\mathbf{C}} = \frac{1}{\mathbf{M}}\mathbf{X}\mathbf{X}^{\mathbf{T}}$.

The output of the linear matched filter for a test input **r**, given the estimated covariance matrix is given by

$$y_r = w^T r = \frac{s^T \hat{C}^{-1} r}{s^T \hat{C}^{-1} s}.$$

In [6] it was shown that using the GLRT the same expression for the linear matched filter can be obtained.

6.2.3.2 Kernel Spectral Matched Filter

Consider the linear model of the input data in a kernel feature space which is equivalent to a non-linear model in the input space

$$\Phi(x) = a_\Phi \Phi(s) + n_\Phi ,$$

where Φ is the non-linear mapping that maps the input data into a kernel feature space, a_Φ is an attenuation constant (abundance measure), the high dimensional vector $\Phi(\mathbf{s})$ contains the spectral signature of the target in the feature space, and vector $\mathbf{n}_\Phi$ contains the added noise in the feature space.

Using the constrained least squares approach that was explained in the previous section it can easily be shown that the equivalent matched filter $\mathbf{w}_\Phi$ in the feature space is given by

$$w_{\Phi} = \frac{\hat{C}_{\Phi}^{-1}\Phi(s)}{\Phi(s)\hat{C}_{\Phi}^{-1}\Phi(s)},$$

where $\hat{\mathbf{C}}_{\mathbf{\Phi}}$ is the estimated covariance of pixels in the feature space. The estimated covariance is given by $\hat{\mathbf{C}}_{\mathbf{\Phi}} = \frac{1}{\mathbf{N}}\mathbf{X}_{\mathbf{\Phi}}\mathbf{X}_{\mathbf{\Phi}}^{\mathbf{T}}$ assuming the sample mean has already been removed from each sample (centered), where $\mathbf{X}_{\mathbf{\Phi}} = [\mathbf{\Phi}(\mathbf{x}_1)\mathbf{\Phi}(\mathbf{x}_2)\cdots\mathbf{\Phi}(\mathbf{x}_{\mathbf{N}})]$ is a matrix whose columns are the mapped input reference data in the feature space. The matched filter in the feature space is equivalent to a non-linear matched filter in the input space and its output for the input $\mathbf{\Phi}(\mathbf{r})$ is given by

$$\mathbf{y}_{\mathbf{\Phi}(\mathbf{r})} = \mathbf{w}_{\mathbf{\Phi}}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{r}) = \frac{\mathbf{\Phi}(\mathbf{s})^{\mathbf{T}}\hat{\mathbf{C}}_{\mathbf{\Phi}}^{-1}\mathbf{\Phi}(\mathbf{r})}{\mathbf{\Phi}(\mathbf{s})^{\mathbf{T}}\hat{\mathbf{C}}_{\mathbf{\Phi}}^{-1}\mathbf{\Phi}(\mathbf{s})}. \tag{24}$$

We now show how to kernelize the matched filter expression equation (24) where the resulting non-linear matched filter is called the kernel matched filter. The pseudo-inverse of the estimated background covariance matrix can be written as

$$\hat{C}_{\Phi}^{\#} = X_{\Phi}\mathrm{B}\Lambda^{-1}\mathrm{B}^{T}X_{\Phi}^{T}. \tag{25}$$

Inserting equation (24) into equation (25) it can be rewritten as

$$\mathbf{y}_{\mathbf{\Phi}(\mathbf{r})} = \frac{\mathbf{\Phi}(\mathbf{s})^{\mathbf{T}}\mathbf{X}_{\mathbf{\Phi}}\mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{X}_{\mathbf{\Phi}}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{r})}{\mathbf{\Phi}(\mathbf{s})\mathbf{X}_{\mathbf{\Phi}}\mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}^{\mathbf{T}}\mathbf{X}_{\mathbf{\Phi}}^{\mathbf{T}}\mathbf{\Phi}(\mathbf{s})}. \tag{26}$$

Also using the properties of the Kernel PCA, we have the relationship

$$\mathbf{K}^{-1} = \mathbf{B}\mathbf{\Lambda}^{-1}\mathbf{B}^{\mathbf{T}}. \tag{27}$$

We denote $\mathbf{K} = \mathbf{K}(\mathbf{X},\mathbf{X}) = (\mathbf{K})_{\mathbf{ij}}$ an N$\times$ N Gram kernel matrix whose entries are the dot products $<\mathbf{\Phi}(\mathbf{x_i}),\mathbf{\Phi}(\mathbf{x_j})>$.

Substituting equation (27) into equation (26) the kernelized version of SMF is given by

$$y_{K_r} = \frac{K(X,s)^{T}K^{-1}K(X,r)}{K(X,s)K^{-1}K(X,s)} = \frac{K_s^{T}K^{-1}K_r}{K_s^{T}K^{-1}K_s}, \tag{28}$$

where the empirical kernel maps $\mathbf{K_s} = (\mathbf{K}(\mathbf{X},\mathbf{s})$ and $\mathbf{K_r} = \mathbf{K}(\mathbf{X},\mathbf{r})$. As in the previous section the kernel matrix **K** as well as the empirical kernel maps needs to be properly centered.

6.3 Implementation of kernel-based detectors

6.3.1 Kernel-RX Detector

In this report, the kernel matrix is estimated locally. For each pixel location in the hyperspectral test image, a dual concentric rectangular window is used to separate a local area into two regions—the inner-window region (IWR) and the outer-window region (OWR), as shown in figure 41; the background kernel matrix $\hat{\mathbf{K}}_{\mathbf{b}}^{-1}$ was obtained locally based on the pixel vectors in the OWR. The test pixel vector **r** was obtained from the IWR. The dual concentric window naturally divides the local area into the potential target region, the IWR, and the background region, the OWR. The size of the IWR was set to enclose targets to be detected. The target size is approximated using the prior knowledge of the range, field of view (FOV), and the dimension of the biggest target in the given data set.

Gaussian RBF kernel $\mathbf{k}(\mathbf{x},\mathbf{y}) = \exp(\frac{-\|\mathbf{x}-\mathbf{y}\|^2}{\mathbf{c}})$ was used to implement the kernel RX-algorithm; the value of c was set experimentally. All the pixel vectors in the test image are first normalized by a constant, which is a maximum value obtained from all the spectral components of the spectral vectors in the corresponding test image, so that the entries of the normalized pixel vectors fit into the interval of spectral values between zero and one. The rescaling of pixel vectors was mainly performed to effectively utilize the dynamic range of Gaussian RBF kernel.

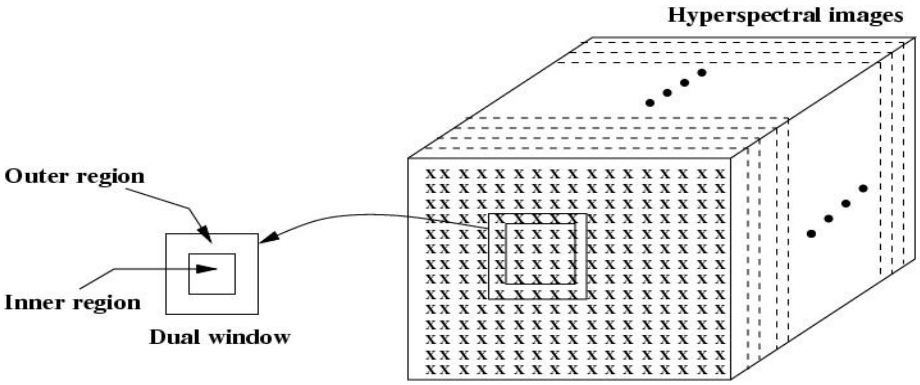


Figure 41. An example of dual concentric window for the kernel-RX anomaly detection.

6.3.2 Kernel Spectral Matched Filter (KSMF)

Background statistics—the kernel matrix **K**—are globally estimated for the kernel-based implementation of the matched filter detectors. The Gaussian RBF kernel is used to implement the KSMF detector. Global estimation must be performed prior to detection and normally needs a large amount of data samples to successfully represent all the background types present in a given data set. In this work, we use a large number of spectral vectors obtained from a given test image that best represent the spectral characteristics of the background. A well-known data clustering algorithm, k-means, is then used on the spectral vectors in order to generate a

significantly less number of spectral vectors (centroids) from which appropriate background statistics are estimated. The number of the representative spectral vectors obtained from the k-means procedure was set to 300 in the experiment. The covariance and kernel matrices were obtained from these 300 spectral vectors. In all the experiments the same target spectral signature **s** is obtained by averaging the target samples collected from one of the targets in the test image.

6.4 Simulation Results

In this section, we apply the proposed detection methods—the polarimetric anomaly detector, the kernel-RX anomaly detector and the kernel spectral matched filter detector (KSMFD)—to hyperspectral images to detect anomalies or materials of interest. We have used hyperspectral test images generated by the two different hyperspectral sensors—SECOTS and DBHSI. The SECOTS generates 66 bands across the whole spectral range ($0.54 - 0.86\,\mu m$) and the DBHSI sensor operates at both the mid-wave infrared (MWIR) and long-wave infrared (LWIR) bands and generates 128 hyperspectral images for each band.

Two AOTF images (AOTF-I and AOTF-II) and a DBHSI image (DBHSI-I) are used as test images, as shown in figure 42. The AOTF-I and AOTF-II images include the chain-link gate and chain-link fence located in the vegetative background, respectively, and the DBHSI image includes a man standing by the chain-link gate on the grass area. The chain-link gate and fence in the images are the targets of interest to be detected.

6.4.1 Polarimetric Anomaly Detection

The size of the local sliding window used for the polarimetric detection was 5x5 pixel areas. For each pixel location, the two groups of the spectral vectors, each from the local window in either of the two polarimetric images, were collected and the Bhattacharyya distance was then measured between the two groups of the spectral vectors.

Figure 43 shows the polarimetric anomaly detection results for the AOTF-I image. The painted metal frame of the chain-link gate in the AOTF-I image was clearly detected while the grass area is highly suppressed. It was possible to detect the metal frame mainly because its surface is painted and specular.

(a) AOTF image I (b) AOTF image II (c) DBHSI image

Figure 42. Sample band images from the AOTF and DBHSI sensors.

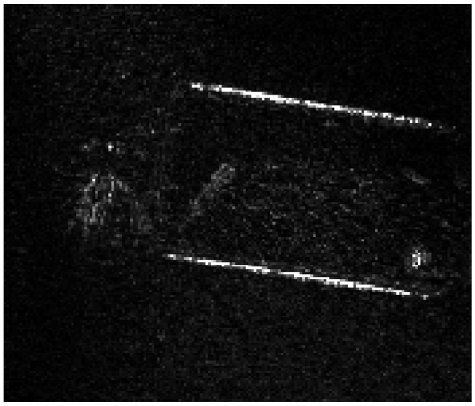
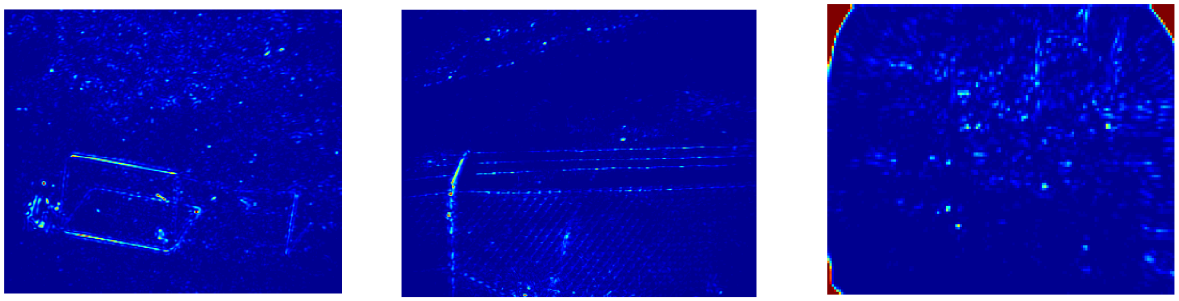


Figure 43. Detection results for the AOTF-I image using the polarimetric anomaly detector.

6.4.2 Kernel-RX Anomaly Detection

In the kernel-RX anomaly detection, the sizes of the IWR and OWR used for the local kernel matrix estimation were 3x3 and 15x15 pixel areas, respectively. The size of the OWR was set to include a sufficient number of the background spectral vectors to generate the kernel matrix. figure 44 shows the detection results for AOTF-I, AOTF-II, and DBHSI-I using the kernel-RX algorithm described in equation (23). Kernel-RX detected the metal frame of the chain-link gate in AOTF-I and the chain-link fence and the barbed-wires in AOTF-II with high contrast in detection values with respect to those of the background. However, for DBHSI-I kernel-RX failed to detect the chain-link gate mainly because of the unstable spectral signatures produced by DBHSI, as shown in figure 44.



(a) AOTF-I (b) AOTF-II (c) DBHSI(LWIR)

Figure 44. Detection results for AOTF-I, AOTF-II, and DBHSI using the kernel-RX algorithm.

6.4.3 Kernel Spectral Matched Filter Detector (KSMFD)

In this section, we apply KSMFD described in equation (28) to the AOTF-I, AOTF-II and DBHSI-I images. The spectral signatures of material of interest are directly obtained from the test images. Figure 45 shows the detection results for AOTF-I using KSMFD. In this experiment, two different target signatures are obtained from the two different locations in the test image—the metal frame and the chain-wire of the gate—and the corresponding detection results were shown in figure 45 (a) and (b), respectively. As can be seen in figure 45, KSMFD with any of the two target signatures clearly detected the chain-link gate with a relatively small number of false alarms.

KSMDF was also applied to the AOTF-II image. For the detection results shown in figure 46 (a) and (b), the target signatures were obtained from the metal frame and the chain-link wire in the AOTF-I image, respectively. For the results in figure 46 (c) and (d) the target signatures were obtained from the metal pole and chain-link wire in the test image AOTF-II, respectively. KSMFD with the target signature obtained from the chain-link wire in AOTF-II detected most of the chain-link fence with a relatively high level of false alarm, as shown in figure 46 (d). KSMFD with the target signature obtained from the metal pole of the chain-link fence in AOTF-II detected the fence fairly well with a relatively low false alarm rate, as shown in figure 46 (c). As can be seen figure 46 (a) and (b), KSMDF with the target signatures obtained from other hyperspectral image taken at a different time and location could still detect the chain-link fence. This is mainly because for the same types of materials (e.g. the metal chain-link gate or chain-link fence) the spectral signatures collected even at a different time and location are not significantly different and KSMFD generally provides good generalization performance.

(a) (b)

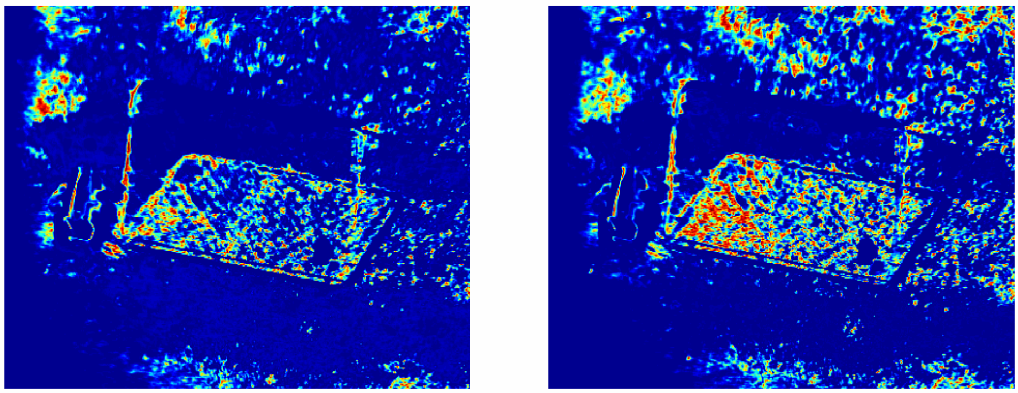


Figure 45. Detection results for AOTF-II. The target signature is obtained from the metal frame in AOTF-I for (a), the chain-link wire in AOTF-I for (b), the metal pole of the chain-link fence in AOTF-II for (c), and the chain-link wire in AOTF-II for (d).

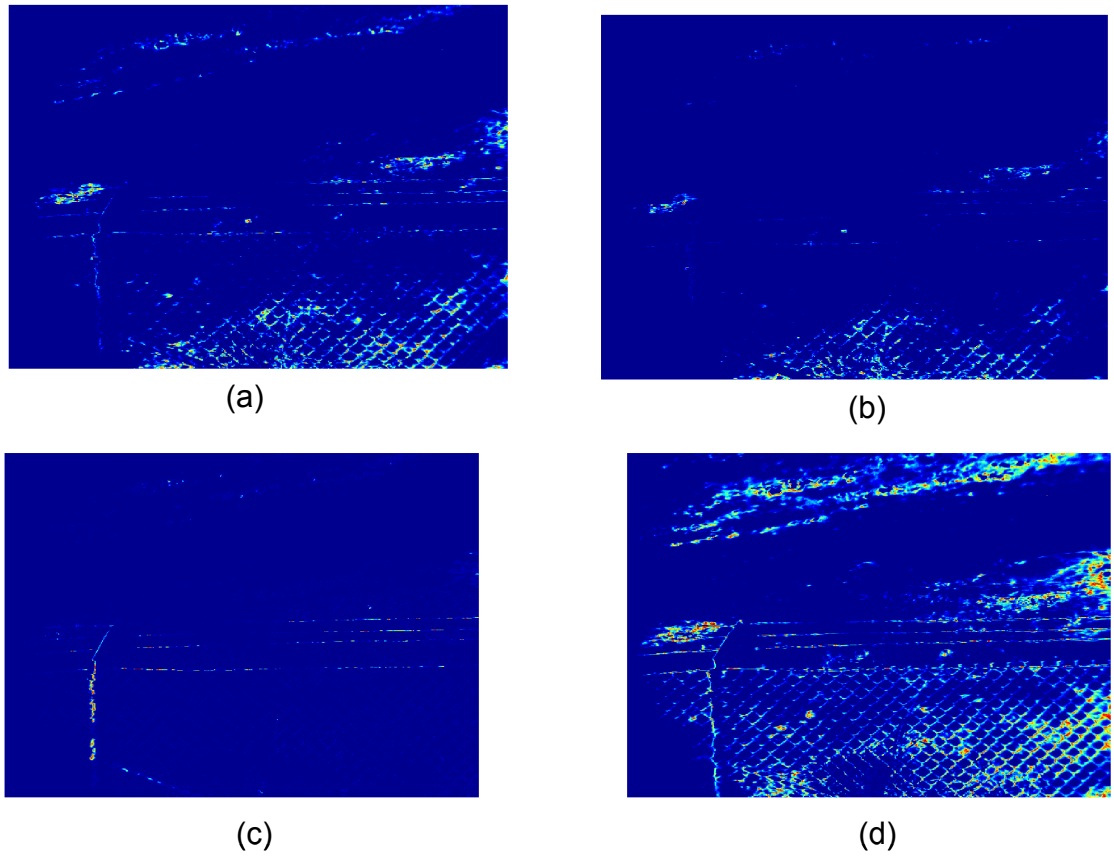


Figure 46. Detection results for AOTF-II. The target signature is obtained from the metal frame in AOTF-I for (a), the chain-link wire in AOTF-I for (b), the metal pole of the chain-link fence in AOTF-II for (c) , and the chain-link wire in AOTF-II for (d).

Figure 47 shows detection results for the DBHSI image using KSMFD. Because the spectral signatures from the DBHSI images are quite unstable and noisy because of the effects of the bad (noisy) pixels on FPA, as shown in figure 48, the detection results using KSMFD are not acceptable, as it detected the frame of the chain-link gate as well as a large portion of the background.

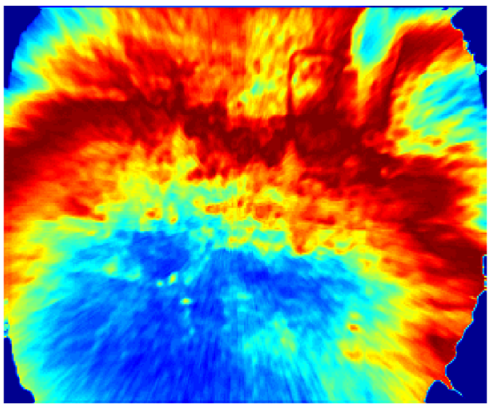
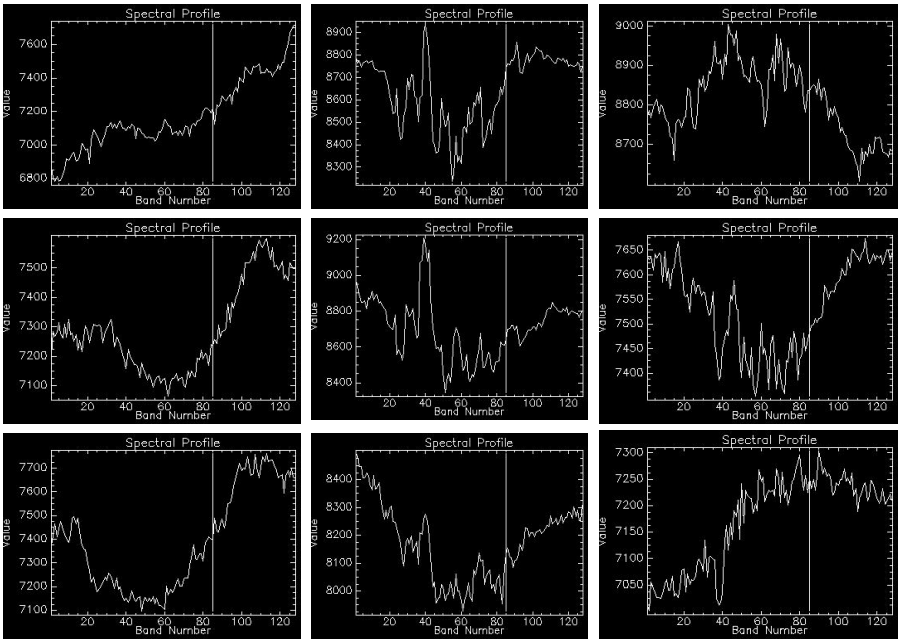


Figure 47. Detection results for DBHSI (LWIR) using KSMD.



(a) (b) (c)

Figure 48. Spectral signatures obtained from three different positions within (a) the man (b) the metal frame of the chain-link gate, and (c) the background.

7. Results

We have applied both anomaly detection and target classification techniques to hyperspectral images to detect objects such as a wire that could obstruct UGV navigation. For anomaly detection, the polarimetric anomaly detector and the kernel-based nonlinear RX detector have been used to detect a chain-link gate or fence. In anomaly detection, detection was performed without referring to the reference spectral signatures of material of interest. Only spectral information contained in the given data set was used to detect an anomaly. In particular, the kernel RX detector—the nonlinear extension of the well-known RX anomaly detector—generated satisfactory performance identifying objects such as the chain-link gate or fence from the background. The improved performance of the kernel RX detector is due mainly to exploitation of the high order correlations of spectral data which the conventional RX detector ignores.

Target classification was also performed, which, unlike anomaly detection, uses a reference spectral signature to detect a target of interest. In this work the kernel-based spectral matched filter detector—the nonlinear extension of the conventional spectral matched filter—was applied to detect a target of interest. The kernel-based matched detector provided enhanced performance, as shown in the experimental results. Especially, even when the reference signature was obtained from the different hyperspectral data collected at a different time and location, the kernel-based matched filter detector was still able to detect the target of interest.

8. Conclusions

In this report, an in-depth study of obstacle detection for autonomous UGV navigation using various hyperspectral imaging sensors has been presented. Three hyperspectral sensors with different operating spectral ranges – DBHSI, SECOTS, and SOC-700 – were used to provide spectral data to the hyperspectral detection algorithms. DBHSI operates at the mid- and long-wave IR bands and the spectral range of SECOTS and SOC-700 spans the visible band. DBHSI is based on an innovative idea of using a diffractive optic lens in combination with a dual-color focal plane array (FPA) to obtain hyperspectral images in two separate IR spectral regions. SECOTS was integrated from commercial off-the-shelf components.

Newly developed hyperspectral anomaly and target detection algorithms have been applied to the hyperspectral images generated by DBHSI, SECOTS, and SOC-700. As shown in the experimental results, in general, the developed algorithms satisfactorily detect objects such as military vehicles, a barbed wire, and a chain-link fence. It can be expected that the developed hyperspectral sensing system will help UGVs navigate an unknown area more safely with increased speed.

However, some areas for improvement in the hyperspectral sensors and detection techniques used in this work remain for future work:

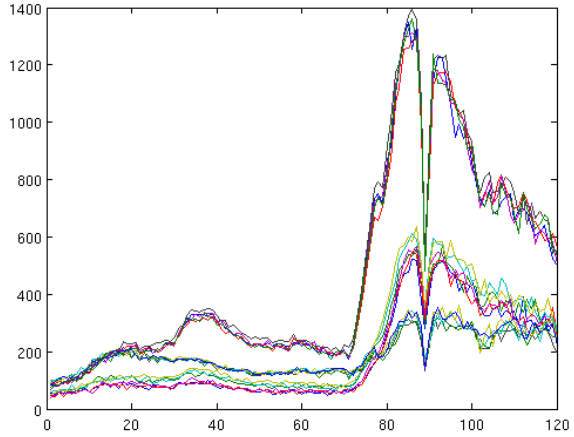
- An improved artifacts removal technique for DBHSI has to be used to mitigate the effects of the bad (noisy) pixels on the deconvolution process; otherwise the deconvoluted images become very noisy, significantly affecting detection performance.
- For SECOTS the image drift should be minimized during data collection; an optical wedge could be integrated.
- Higher collection rates are needed to help reduce effects of the time varying illumination for the hyperspectral sensors operating at the visible band.
- A reduction in data processing time of the hyperspectral detection algorithms is needed.
- Construction of a spectral library for materials of interest for the hyperspectral target classification will help identify potentially harmful objects for UGV navigation.
- Band selection is needed to identify the minimum number of spectral bands given specific applications and hyperspectral sensors.

Hyperspectral band selection is a particularly important issue because a large number of spectral bands generated by hyperspectral sensors often cause a high level of computational complexity and require large storage space. Effective band selection techniques will remove redundant spectral information and choose specific bands that contribute most to specific tasks.

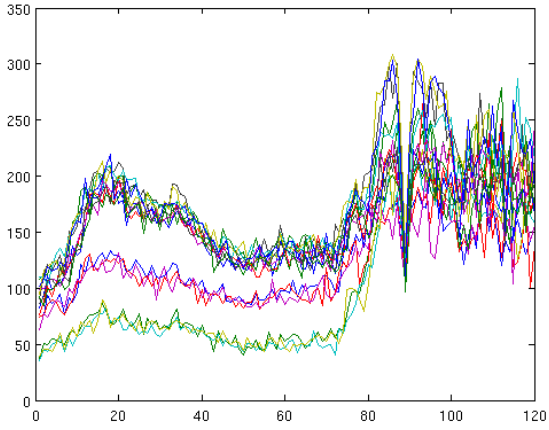
9. References

1. Goldberg, Arnold C., et al, "FOCAL PLANE ARRAYS—Comparison of HgCdTe and quantum-well infrared photodetector dual-band focal plane arrays," Opt. Eng. **42**, no. 1, p. 30–46, 2003.
2. Hinnrichs, M.; Gupta, N.; and Goldberg, A. "Dual Band (MWIR/LWIR) Hyperspectral Imager," Proc. of AIPR Workshop, p. 73–78, 2003.
3. HyPAT II User's Manual, version 4.04 from The Pacific Advance Technology, Buellton, CA.
4. Rosario, D. Innovative Statistical Inference for Anomaly Detection in Hyperspectral Imagery, ARL Technical Report ARL-TR-3339, September 2004.
5. Qin, J. and Zhang, B. "A goodness of fit test for logistic regression models based on case-control data," Biometrika, vol 84, 609-618, 1997.
6. Surface Optics Corporation's official website: www.surfaceoptics.com.
7. Yu, X.; Hoff, L.; Reed, I.; Chen, A.; and Stotts, L. "Automatic target detection and recognition in multiband imagery: A unified ML detection and estimation approach," IEEE Tran. Image Processing, vol. 6, pp. 143-156, Jan. 1997.
8. Rosario, D. "Highly effective logistic regression model for signal (anomaly) detection," 2004 IEEE ICASSP Proceedings, May 17-21, 2004.
9. Kwon, H. and Nasrabadi, N. M. Kernel-RX algorithm: A nonlinear anomaly detector for hyperspectral imagery, IEEE Trans. Geosci. Remote Sensing, 43(2):388-397, Feb. 2005.
10. Scholkopf, B. and Smola, A. J. Learning with kernels, Cambridge, MA: The MIT Press, 2002.
11. Reed, I. S. and Yu, X. Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution, IEEE Trans. Acoust., Speech Signal Process., vol. 38, no. 10, pp 1760-1770, Oct. 1990.
12. Scholkopf, B.; Smola, A. J.; and Muller, K.-R. Kernel principal component analysis, Neural Comput., no. 10, pp. 1299-1319, 1999
13. Harsanyi, J. C. Detection and Classification of subpixel spectral signatures in hyperspectral image sequence, Ph.D. dissertation, Dept. Elect. Eng., Univ. of Maryland, Baltimore Co., 1993.
14. Kraut, S.; Scharf, L. L.; and McWhorter T. Adaptive subspace detector, IEEE Trans. Signal Process. 49(1):208-216, Jan 2001.

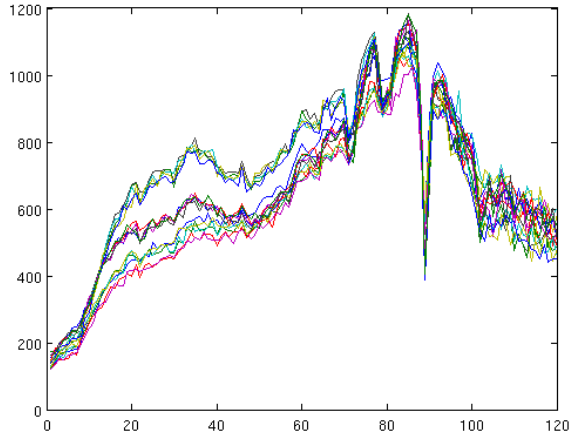
Appendix A.—Samples of each of the 21 classes extracted for training examples.



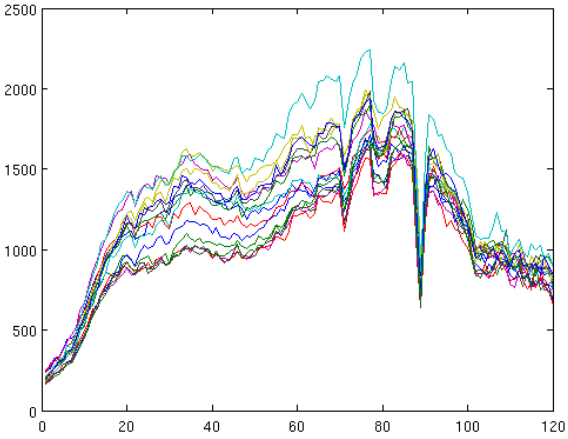
Class 1: tree-leaves



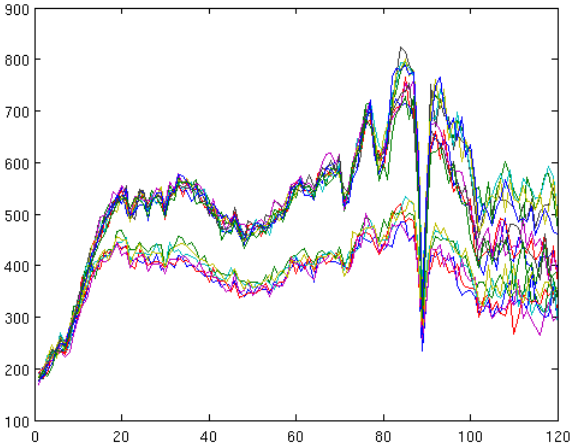
Class 2: tree-bark



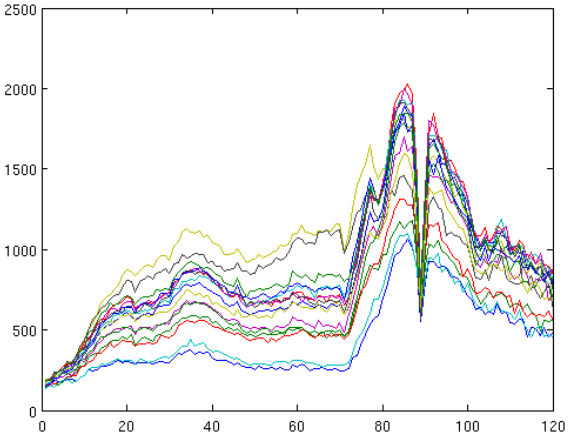
Class 3: grass-brown



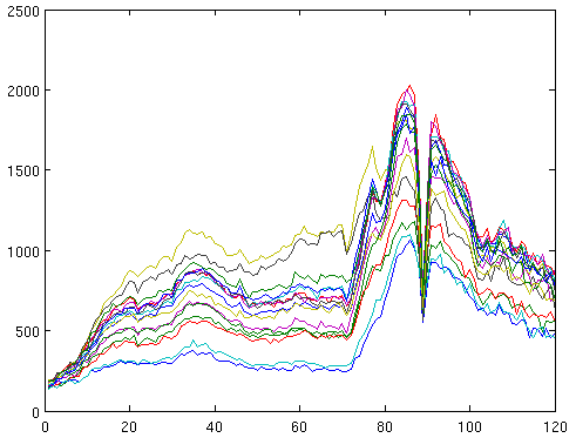
Class 4: grass-yellow



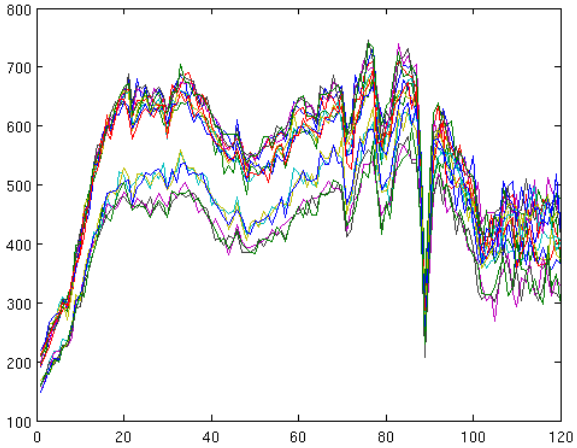
Class 5: Grass-gray



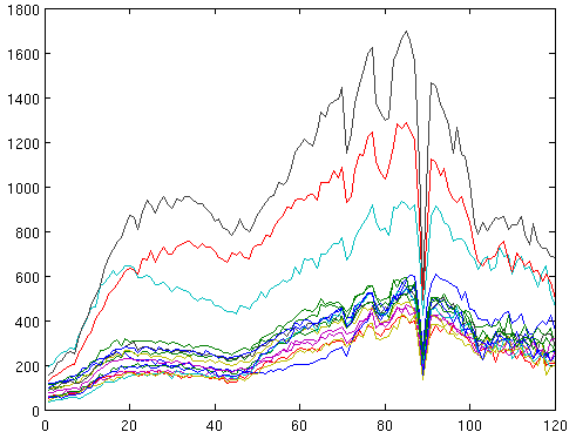
Class 6: Grass-green



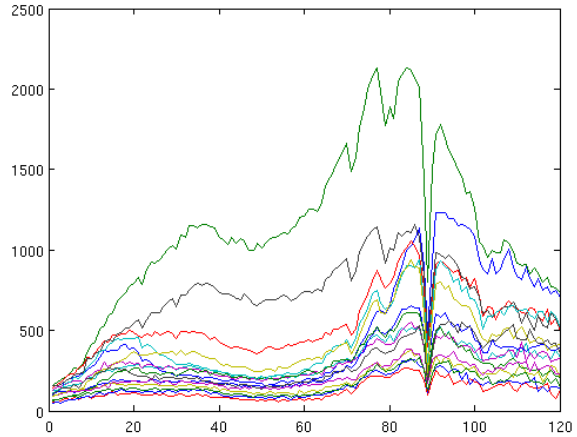
Class 7: dirt



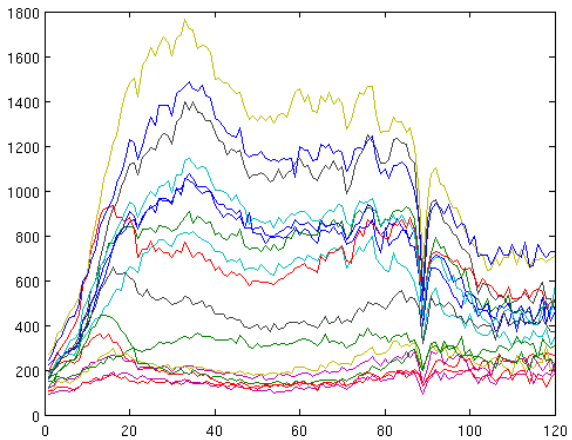
Class 8: Macadam road



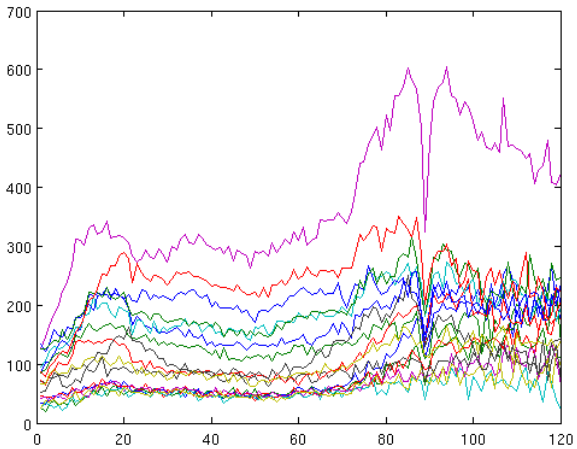
Class 9: Skin-2



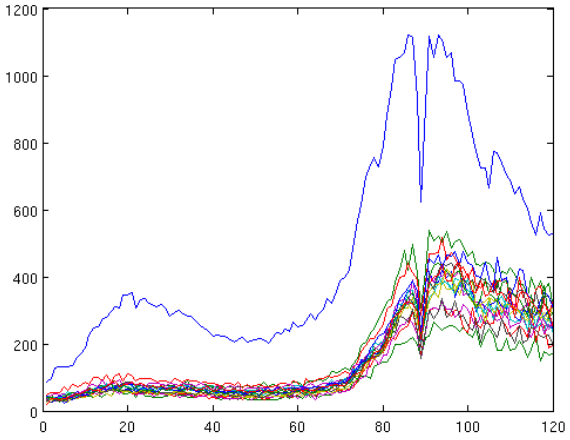
Class10: hat-green



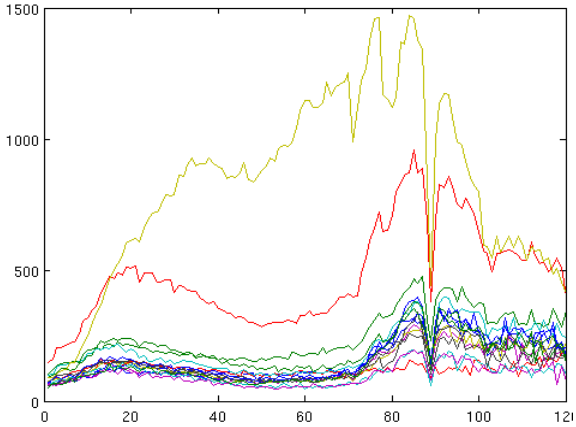
Class 11: hat-blue



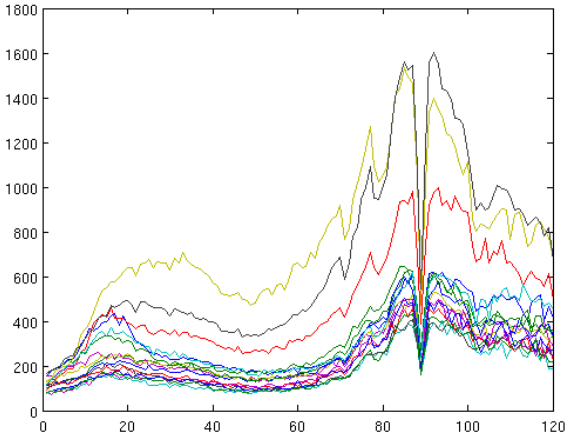
Class 12: Pants-dark



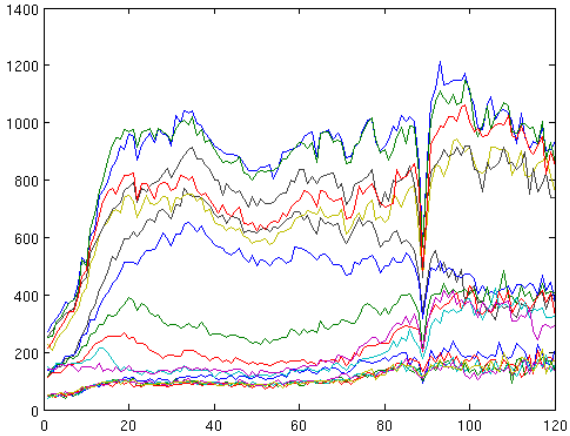
Class 13: shirt-dark



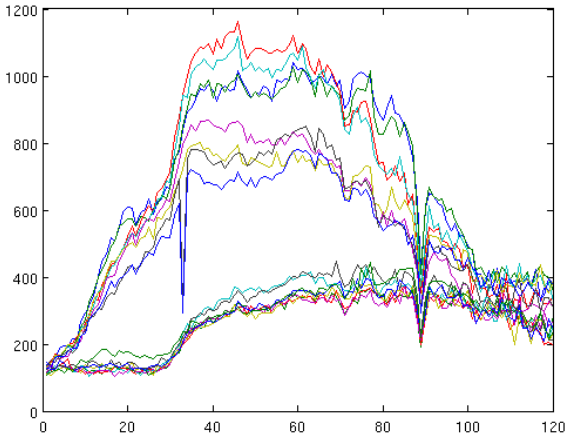
Class14: blue-jeans



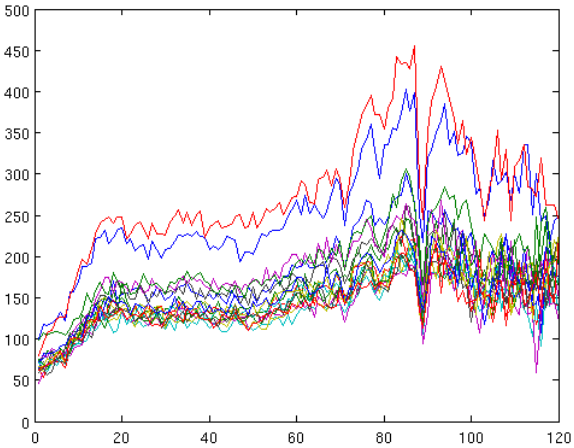
Class 15: shirt-blue



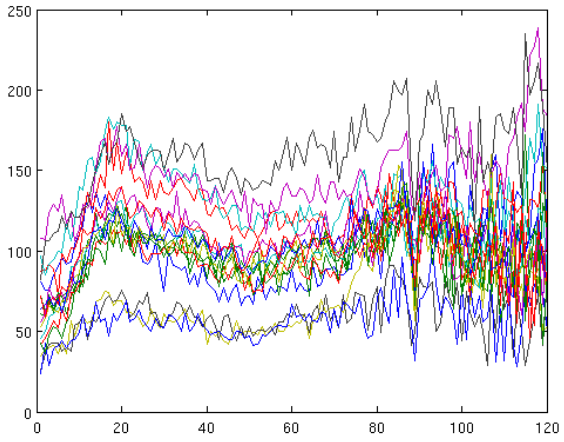
Class 16: stinger-tube



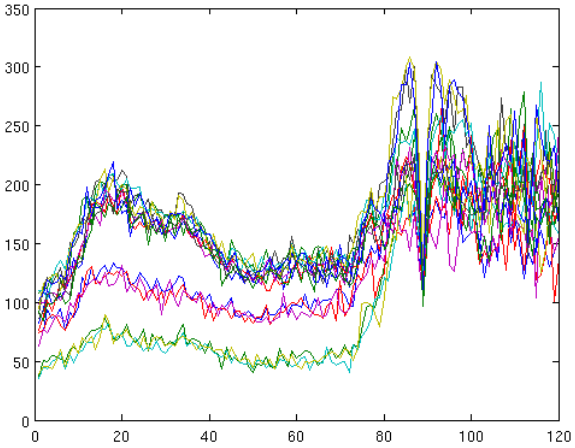
Class 17: stinger-yellow dot



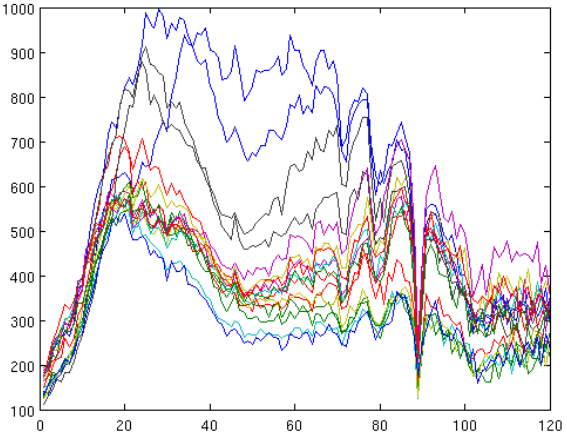
Class 18: boot-brown



Class 19: black-vinyl

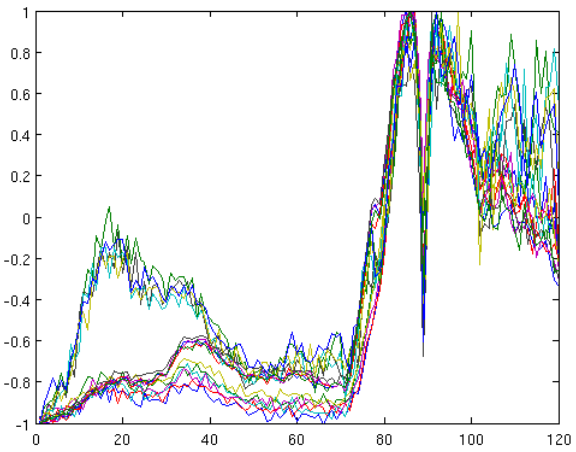


Class 20: pick-up white

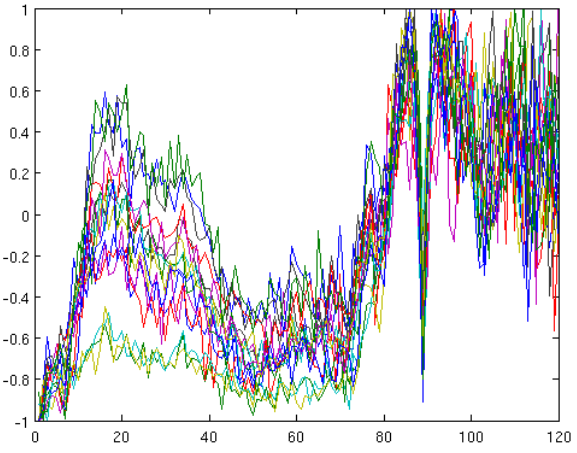


Class 21: pick-up blue

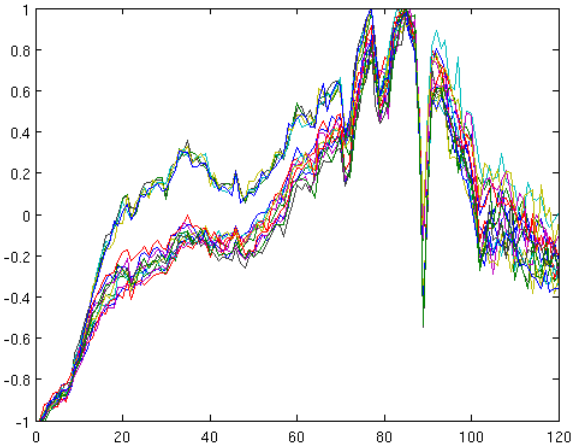
Appendix B.—Normalized training data for each of the 21 classes used to train the neural networks.



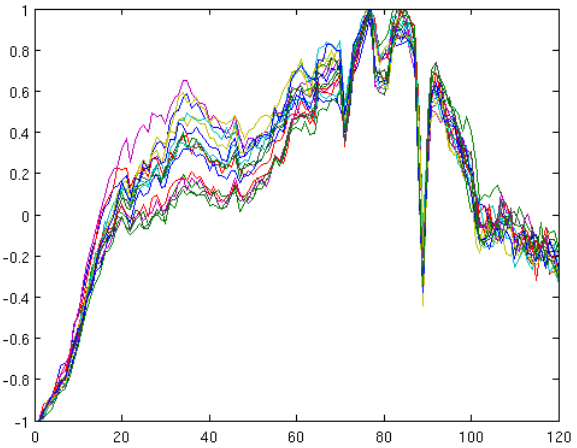
Class 1: tree-leaves



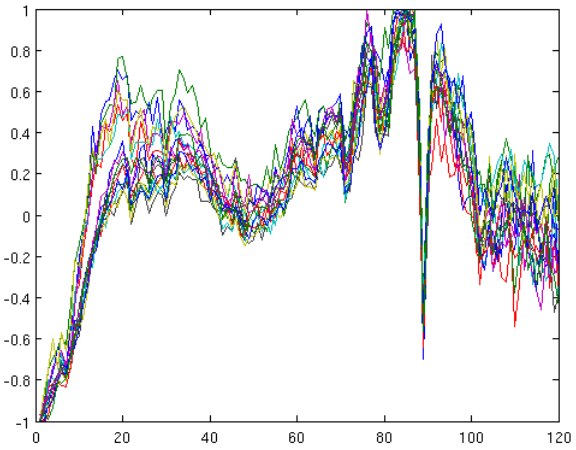
Class 2: tree-bark



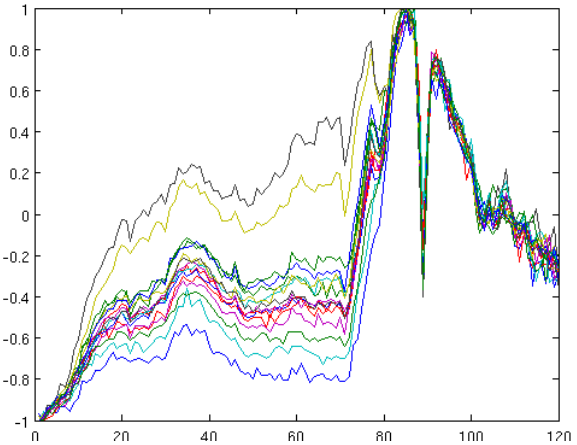
Class 3: grass-brown



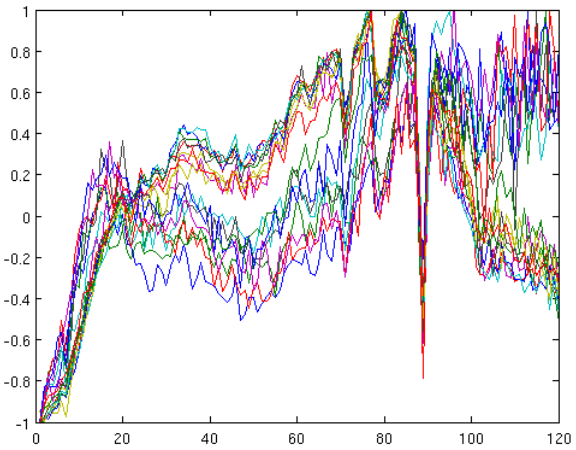
Class 4: grass-yellow



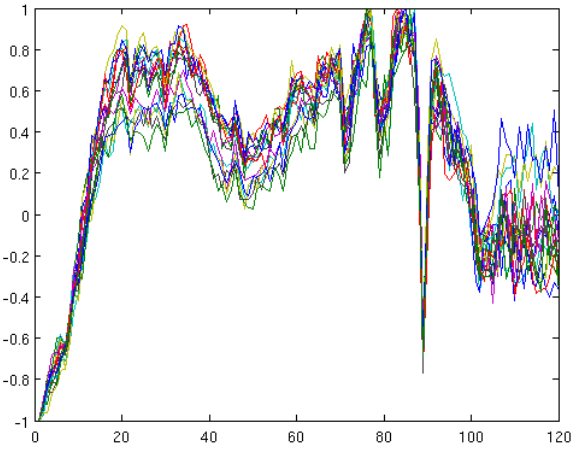
Class 5: grass-gray



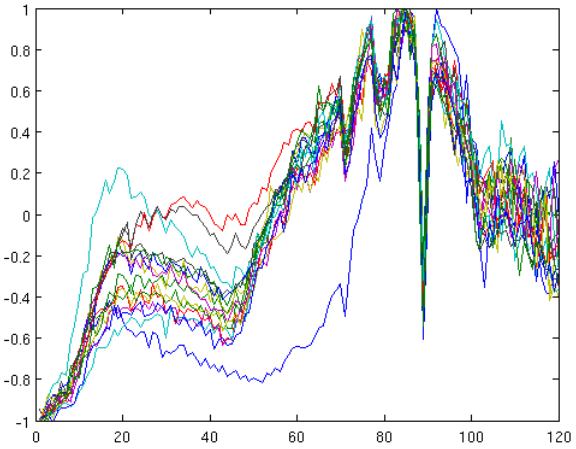
Class 6: grass-green



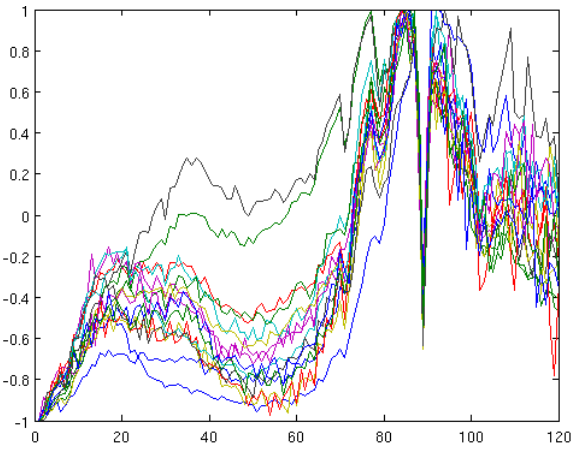
Class 7: dirt



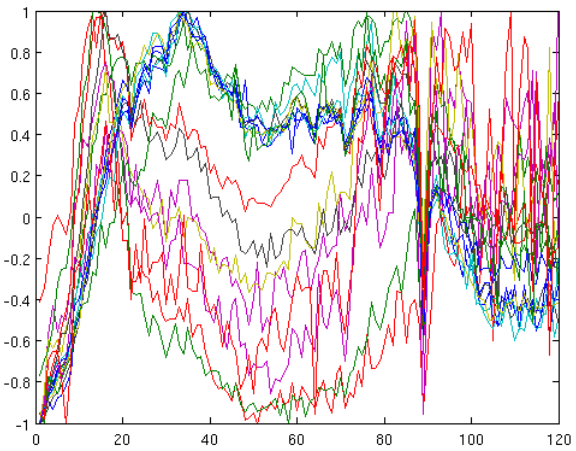
Class 8: macadam road



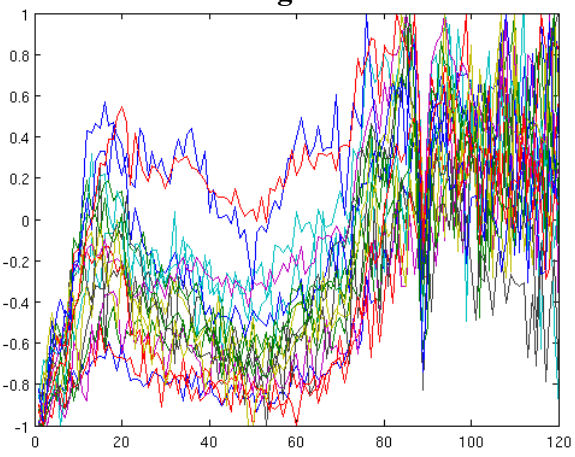
Class 9: skin-2



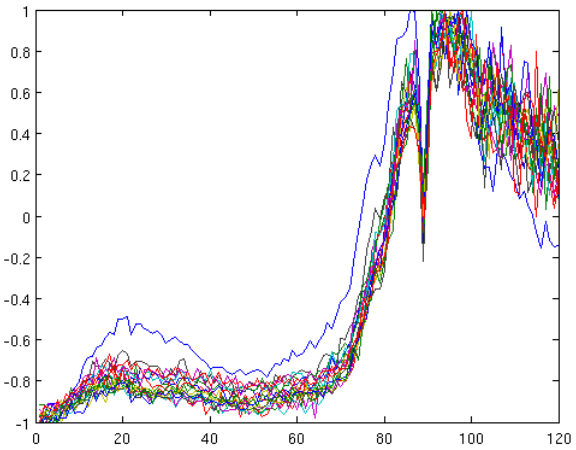
Class 10: hat-green



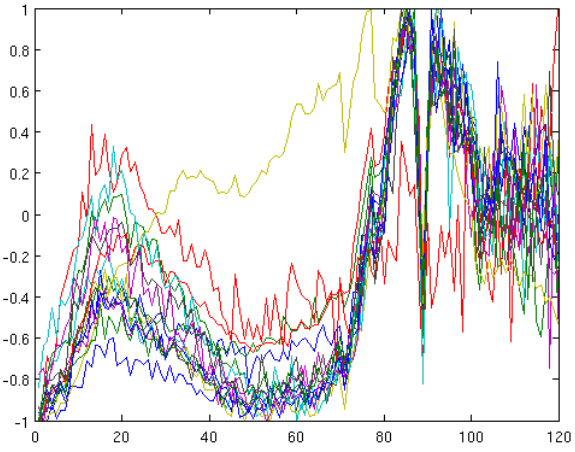
Class 11: hat-blue



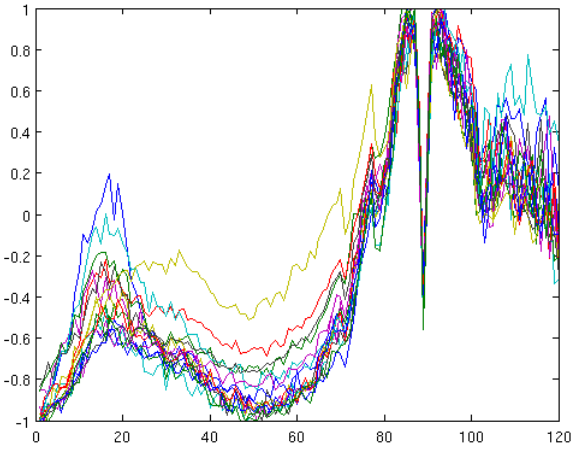
Class 12: pants-dark



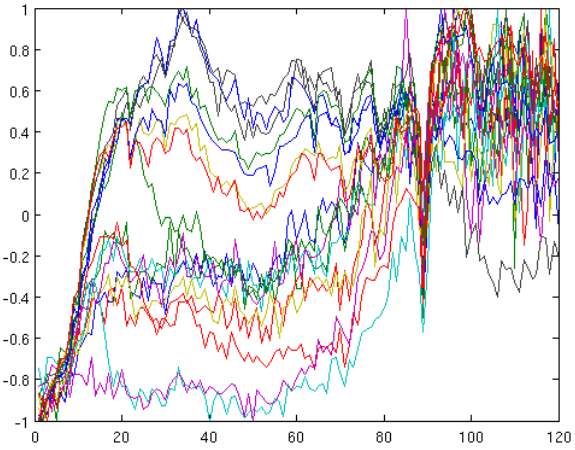
Class 13: shirt-dark



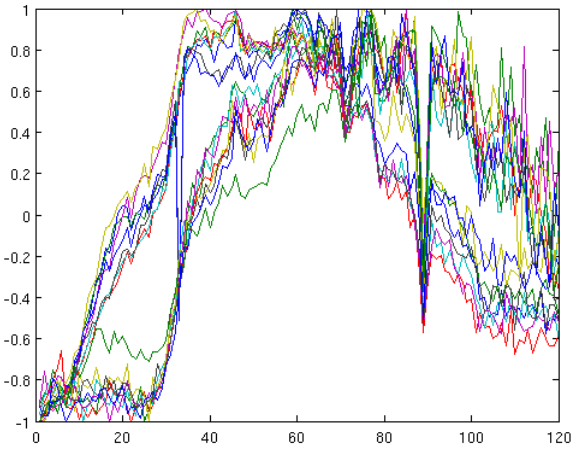
Class 14: blue-jeans



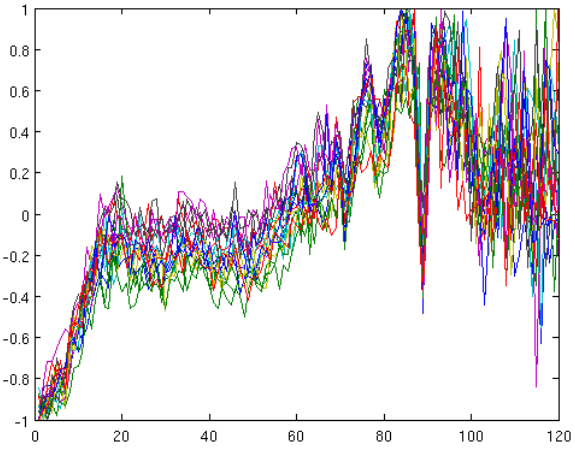
Class 15: shirt-blue



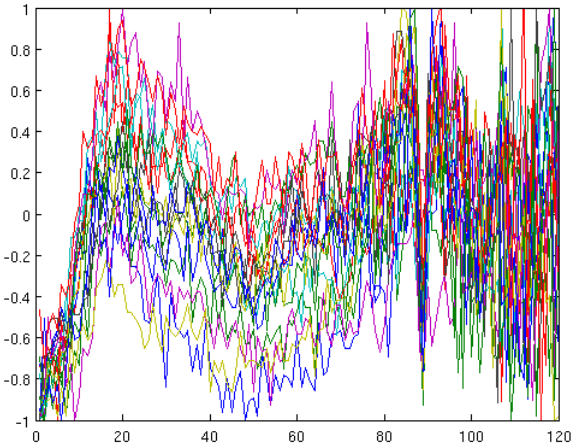
Class 16: stinger-tube



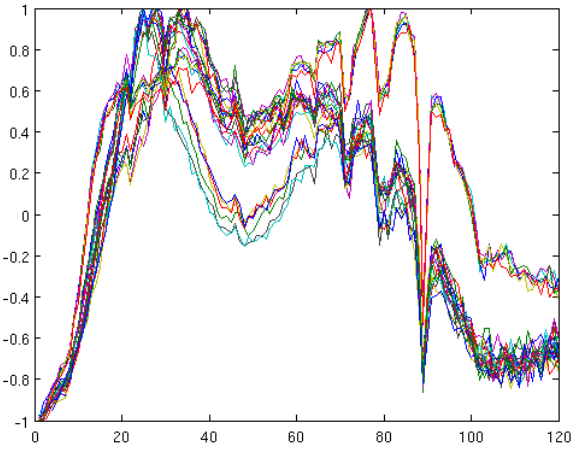
Class 17: stinger-yellow dot



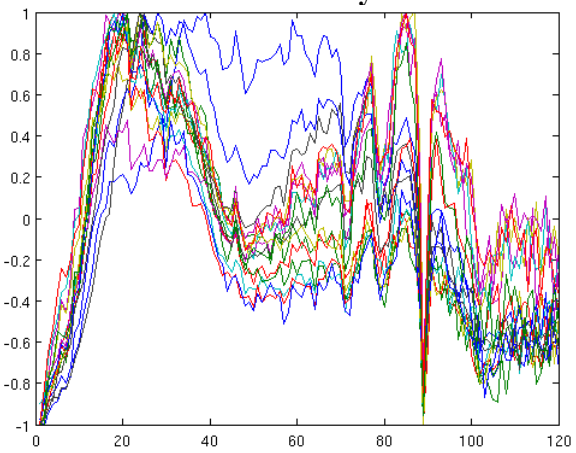
Class 18: boot-brown



Class 19: black-vinyl



Class 20: pick-up white



Class 21: pick-up blue

Distribution List

Admnstr
Defns Techl Info Ctr
ATTN DTIC-OCP (Electronic copy)
8725 John J Kingman Rd Ste 0944
FT Belvoir VA 22060-6218

DARPA
ATTN C Schwartz
ATTN IXO S Welby
ATTN R Hummell
3701 N Fairfax Dr
Arlington VA 22203-1714

Ofc of the Secy of Defns
ATTN ODDRE (R&AT)
The Pentagon
Washington DC 20301-3080

US Army RDECOM AMRDEC
ATTN ATCD-B
FT Monroe VA 23651-5850

Army Rsrch Physics Div
ATTN AMSRD-ARL-RO-MM R Launer
PO Box 12211
Research Triangle Park NC 27709-2211

US Army TRADOC
Battle Lab Integration & Techl Dirctrt
ATTN ATCH-B
10 Whistler Lane
FT Monroe VA 23651-5850

CECOM NVESD
ATTN AMSRD-CER-NV-OD J Ratches
10221 Burbeck Rd Ste 430
FT Belvoir VA 22060-5806

US Army Aberdeen Test Center
ATTN CSTE-DT-AT-WC-A F Carlen
ATTN CSTE-DTC-AT-TC-N D L Jennings
400 Colleran Road
Aberdeen Proving Ground MD 21005-5059

US Army ARDEC
ATTN AMSTA-AR-TD
Bldg 1
Picatinny Arsenal NJ 07806-5000

US Army Aviation & Mis Lab
ATTN AMSRD-AMR-SG-IP H F Anderson
Bldg 5400
Redstone Arsenal AL 35809

US Army CERDEC, NVESD
ATTN AMSRD-CER-NV-xx J Hilger
ATTN AMSRD-CER-NV-xx P Perconti
ATTN AMSRD-CER-NV-xx R Driggers
10221 Burbeck Rd Ste 430
FT Belvoir VA 22060-5806

US Army ERDC
ATTN CEERD-TR-S R Rand
7701 Telegraph Rd
Bldg 2592
Alexandria VA 22315

US Army Materiel Sys Anal Actvty
ATTN AMSRD-AMS-SC E Grove
392 Hopkins Rd
Aberdeen Proving Ground MD 21005-5071

US Army Materiel Sys Anal Actvty
ATTN AMSRD-AMS-SC G Kistner
Aberdeen Proving Ground MD 21005-5071

US Army Materiel Sys Anal Actvty
ATTN AMSRD-AMS-SC J Mazz
392 Hopkins Rd
Aberdeen Proving Ground MD 21005-5071

US Army Natick RDEC
Acting Techl Dir
ATTN SBCN-TP P Brandler
Kansas Street Bldg 78
Natick MA 01760-5056

US Army RDECOM AMRDEC
ATTN ATCD-B
FT Monroe VA 23651-5850

US Army RDECOM AMRDEC
ATTN AMSRD-AMR-SG-IP R Sims
Bldg 5400
Redstone Arsenal AL 35898

US Army RDECOM AMRDEC
ATTN AMSRD-AMR-WS-PL W Davenport
Bldg 7804
Redstone Arsenal AL 35898

Commanding General
US Army RDECOM AMRDEC
ATTN AMSRD-AMR W C McCorkle
Bldg 5400
Redstone Arsenal AL 35898-5000

US Army RDECOM ARDEC
ATTN AMSRD-AAR-QES P Willson
Radiographic Laboratory, B.908
Picatinney Arsenal NJ 07806-5000

US Army RDECOM TARDEC
ATTN AMSRD-TAR-R G R Gerhart
MS 263
Warren MI 48397-5000

US Army Soldier & Biological Chem Ctr
ATTN AMSSB-RRT-DP W Loerop
Edgewood Chem & Biological Ctr
Bldg E-5554
Aberdeen Proving Ground MD 21010-5424

Commander
USAISEC
ATTN AMSEL-TD Blau
Building 61801
FT Huachuca AZ 85613-5300

AFRL/SNAA
ATTN M Jarratt
2241 avionics Circle Area B, Bldg 620
Wright Patterson AFB OH 45433-7321

CMTCO
ATTN MAJ A Suzuki
1030 S Highway A1A
Patrick AFB FL 23925-3002

SITAC
ATTN H Stiles
ATTN K White
ATTN R Downie
11981 Lee Jackson Memorial Hwy Suite 500
Fairfax VA 22033-3309

US Army Rsrch Lab
ATTN AMSRD-ARL-WM-BF G Haas
ATTN AMSRD-ARL-WM-BF W Oberle
Aberdeen Proving Ground MD 21005-5067

Director
US Army Rsrch Lab
ATTN AMSRD-ARL-RO-D JCI Chang
PO Box 12211
Research Triangle Park NC 27709

US Army Rsrch Lab
ATTN AMSRD-ARL-RO-EL W Sander
PO Box 12211
Research Triangle Park NC 27709-2211

US Army Rsrch Office
ATTN AMSRD-ARL-RO-PP R Hammond
PO box 12211
Research Triangle Park NC 27709-2211

US Army Rsrch Lab
ATTN AMSRD-ARL-CI-OK-T Techl Pub (2 copies)
ATTN AMSRD-ARL-CI-OK-TL Techl Lib (2 copies)
ATTN AMSRD-ARL-D A Grum
ATTN AMSRD-ARL-D J M Miller
ATTN AMSRD-ARL-D J Rocchio
ATTN AMSRD-ARL-SE J Pellegrino
ATTN AMSRD-ARL-SE-S J Eicke
ATTN AMSRD-ARL-SE-SE H Kwon (10 copies)
ATTN AMSRD-ARL-SE-SE N Nasrabadi
ATTN AMSRD-ARL-SE-SE P Gillespie
ATTN IMNE-ALC-IMS Mail & Records Mgmt