High-Resolution, Slant-Angle Scene Generation and Validation of Concealed Targets in DIRSIG

by

Captain Kris Barcomb, USAF

B.S. Clarkson University, 1999
M.B.A. University of La Verne, 2002

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

August 24, 2004

Signature of the Author _____

Accepted by _____
Coordinator, M.S. Degree Program Date

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
27 SEP 2004	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
High-Resolution, Slant-Angle Scene Generation and Validation of Concealed Targets in DIRSIG	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Rochester Institute of Technology**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	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
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**177**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

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

CERTIFICATE OF APPROVAL

M.S. DEGREE THESIS

The M.S. Degree Thesis of Captain Kris Barcomb, USAF
has been examined and approved by the
thesis committee as satisfactory for the
thesis required for the
M.S. degree in Imaging Science

Dr. John Schott, Thesis Advisor

Dr. Carl Salvaggio

Scott Brown

Date

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

Title of Thesis:

High-Resolution, Slant-Angle Scene Generation and Validation of Concealed Targets in DIRSIG

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

Signature _____

Date

High-Resolution, Slant-Angle Scene Generation and Validation of Concealed Targets in DIRSIG

by

Captain Kris Barcomb, USAF

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

Abstract

Traditionally, synthetic imagery has been constructed to simulate images captured with low resolution, nadir-viewing sensors. Advances in sensor design have driven a need to simulate scenes not only at higher resolutions but also from oblique view angles. The primary efforts of this research include: real image capture, scene construction and modeling, and validation of the synthetic imagery in the reflective portion of the spectrum. High resolution imagery was collected of an area named MicroScene at the Rochester Institute of Technology using the Chester F. Carlson Center for Imaging Science's MISI and WASP sensors using an oblique view angle. Three Humvees, the primary targets, were placed in the scene under three different levels of concealment. Following the collection, a synthetic replica of the scene was constructed and then rendered with the Digital Imaging and Remote Sensing Image Generation (DIRSIG) model configured to recreate the scene both spatially and spectrally based on actual sensor characteristics. Finally, a validation of the synthetic imagery against the real images of MicroScene was accomplished using a combination of qualitative analysis, Gaussian maximum likelihood classification, grey-level co-occurrence matrix derived

texture metrics, and the RX algorithm. The model was updated following each validation using a cyclical development approach. The purpose of this research is to provide a level of confidence in the synthetic imagery produced by DIRSIG so that it can be used to train and develop algorithms for real world concealed target detection.

Disclaimer

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Contents

1 Introduction **1**

2 Background **7**

2.1 Synthetic Image Generation 7

2.1.1 Physical Miniature Models 9

2.1.2 Infrared Modeling and Analysis (Irma) 10

2.1.3 Camouflage Electro-Optic Simulation System (CAMEO-SIM) . . 12

2.1.4 Visual and Electro-Optical (VISEO) Detection Analysis System . 15

2.1.5 Digital Imaging and Remote Sensing Image Generation (DIRSIG) 18

2.2 Sensors 30

2.2.1 Multispectral Imaging Spectrometer Instrument (MISI) 30

2.2.2 Wildfire Airborne Sensor Program (WASP) 34

2.3 Verification and Validation 35

2.3.1 Rank-Order Correlation (ROC) 36

2.3.2 Gaussian Maximum Likelihood Classification 38

2.3.3 Gray-Level Co-Occurrence Matrices (GLCM) 39

2.3.4 RX 45

2.3.5 Other Validation Efforts 48

2.4 Camouflage 54
2.4.1 Camouflage Overview 54
2.4.2 Vehicle Camouflage 55
2.4.3 Camouflage in DIRSIG 57

3 Approach **59**
3.1 The Experiment 60
3.1.1 Collection Overview 60
3.1.2 Pre-Collection Phase 66
3.1.3 Data Collection 67
3.2 Virtual MicroScene Construction 71
3.2.1 Synthetic Terrain Creation 72
3.2.2 Terrain Vegetation 76
3.2.3 3-D Models of Manmade Objects 88
3.2.4 Sensor Location and Configurations 95
3.2.5 MISI Image Calibration 97
3.2.6 MISI Point Spread Function 99
3.2.7 MISI Synthetic Noise Generation 101
3.2.8 RX Algorithm Implementation 104

4 Results **108**
4.1 Qualitative Image Comparison 109
4.2 Qualitative Spectral Comparison 114
4.2.1 Control Target Analysis 114
4.2.2 Vegetation and Terrain Analysis 119
4.2.3 Humvee and Small Target Analysis 121

4.2.4 Rank-Order Correlation Results 122
4.3 Gaussian Maximum Likelihood Classification Results 126
4.4 Grey-Level Co-occurrence Matrices Results 131
4.5 RX Results 138

5 Conclusions and Recommendations **146**

A RX Code **158**

B TreePlanter Code **162**

List of Figures

1.1 Organization Chart for Army MURI 2

1.2 WASP image of MicroScene area at Rochester Institute of Technology taken in the fall. 4

2.1 A wire-frame representation of a single DIRSIG object used to construct scenes (Brown, 2003a). 19

2.2 A ray tracer is utilized to determine which objects contribute to the radiance for a given pixel (Brown, 2003a). 20

2.3 Example rendering of the effects of oversampling 21

2.4 Solar and Self-Emitted Photon Paths (Schott, 1997, p. 114) 24

2.5 Simple example of object texture projection 27

2.6 Projecting 2-D texture onto 3-D coordinates 27

2.7 An example of texturing in a beta version of DIRSIG 4. 28

2.8 MISI Noise Correlation Matrix 32

2.9 Illustration of ESF and LSF 34

2.10 4 x 4 image window with four gray tones 41

2.11 General form of a GLCM for gray levels 0 - 3 41

2.12 Horizontal GLCM 42

2.13 Vertical GLCM 42
2.14 Negative Diagonal GLCM 42
2.15 Positive Diagonal GLCM 42
2.16 Diagram of DIRSIG's treatment of NULL material mappings. 58

3.1 Sensor and target locations in MicroScene. 60
3.2 WASP and MISI positioned on a scissor cart. 61
3.3 Humvee target 62
3.4 Humvee target under camouflage. 63
3.5 Complex shadows cast by the camouflage netting. 64
3.6 Terrapix image taken at 1000. 69
3.7 Color image from MISI's vis spectrometer taken at 1000. 69
3.8 Single band from MISI's NIR spectrometer taken at 1800. 70
3.9 Transition region in synthetic MicroScene. 74
3.10 Top: Grass model rendered in DIRSIG – Bottom: Digital image of MicroScene 79
3.11 Blender generated grass model rendered in DIRSIG. 80
3.12 Spatial comparison of WASP and DIRSIG overhead imagery. 84
3.13 TreePro coniferous tree modeled at the branch level. 85
3.14 TreePro coniferous tree modeled at the pine needle level. 87
3.15 Various detailed models built for the virtual MicroScene. 89
3.16 Humvee with support structure rendered in Rhino. 90
3.17 DIRSIG preview image of camouflage netting with non-coplanar quads. 92
3.18 Threshold generated material map for camouflage netting. 93
3.19 Truth vs. DIRSIG imagery comparison of a Humvee underneath camouflage netting. 94

3.20 Transition region used to determine MISI LSF 100
3.21 MISI PSF kernel 101
3.22 Process flow for creating correlated noise for DIRSIG imagery. 103
3.23 RX GUI windows. 105
3.24 Neighborhood and spatial shape of RX kernel 106

4.1 True color MISI vs. DIRSIG image comparison. 110
4.2 .811 μm band MISI vs. DIRSIG image comparison. 111
4.3 MISI vs. DIRSIG spectral comparison in the visible region at 1000. . . . 115
4.4 MISI vs. DIRSIG spectral comparison in the visible region at 1000. . . . 116
4.5 MISI vs. DIRSIG spectral comparison in the NIR region at 1800. 117
4.6 MISI vs. DIRSIG spectral comparison in the NIR region at 1800. 118
4.7 ROC results for .458μm band. 122
4.8 ROC results for .546μm band. 123
4.9 ROC results for .640μm band. 123
4.10 ROC results for .731μm band. 124
4.11 ROC results for .865μm band. 124
4.12 ROC results for .993μm band. 125
4.13 1000 MISI vs. DIRSIG GML classification comparison. 127
4.14 1800 MISI vs. DIRSIG GML classification comparison. 130
4.15 Comparison of GLCM results with a 1 pixel shift in the x direction using a 31x31 window on the .555μm band in the 1000 imagery. 132
4.16 Comparison of GLCM results with a 1 pixel shift in the y direction using a 31x31 window on the .555μm band in the 1000 imagery. 133
4.17 Comparison of GLCM results with a 1 pixel shift in the x direction using a 31x31 window on the .776μm band in the 1800 imagery. 133

4.18 Comparison of GLCM results with a 1 pixel shift in the y direction using a 31x31 window on the .776μm band in the 1800 imagery. 133
4.19 Mean GLCM values of grass regions : 1=Homogeneity, 2=Contrast, 3=Entropy. 134
4.20 Mean GLCM values of trees regions : 1=Homogeneity, 2=Contrast, 3=Entropy. 135
4.21 Mean GLCM values of Dirt regions : 1=Homogeneity, 2=Contrast, 3=Entropy. 135
4.22 GML classification of 1000 imagery that includes the derived GLCM images. 137
4.23 RX run in anomaly detection mode on 1000 RX imagery. 138
4.24 RX run in anomaly detection mode on 1800 RX imagery. 139
4.25 RX run in matched filter mode using DIRSIG derived Humvee on 1000 imagery. 141
4.26 RX run in matched filter mode using DIRSIG derived Humvee on 1800 imagery. 141
4.27 Histograms of the RX algorithm results from the 1000 imagery. 143
4.28 Histograms of the RX algorithm results from the 1800 imagery. 144

List of Tables

2.1 Summary of variable meanings and corresponding photon paths 25

3.1 Summary of command line options for TreePlanter.pro. 82

Chapter 1

Introduction

The purpose of the research is to perform verification and validation (V&V) of synthetic images of concealed vehicles created using the Digital Imaging and Remote Sensing (DIRS) Lab's Digital Imaging and Remote Sensing Image Generation (DIRSIG) model. This work includes ground truth and truth image collection of the MicroScene area at the Rochester Institute of Technology (RIT), synthetic scene development based on that area, and the verification and validation of the synthetic scene as compared to the truth imagery. The funding for the Multi-University Research Initiative (MURI) comes from the Army Research Organization (ARO). This research is a portion of the overall objective of the MURI. Figure 1.1 shows the interaction and responsibilities of the various schools that are participating in the research. The ultimate goal of the MURI project is to provide training sets and development environments based on synthetic imagery for automatic target recognition (ATR) algorithms. This thesis will explain why synthetic imagery is important, compare various approaches to synthetic scene generation, give an overview of the methods and algorithms to be used to validate the results, and explain how the research was conducted.

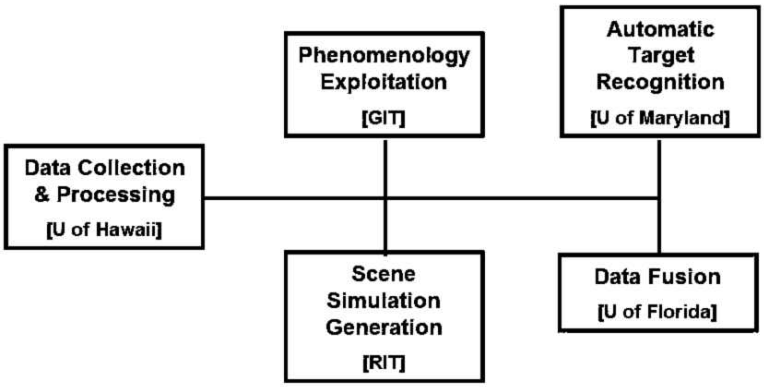


Figure 1.1: Organization Chart for Army MURI

Many challenges were faced over the course of this work. One challenge was using DIRSIG to model scenes at very high spatial resolution. As mentioned, the ultimate goal of the Army's project is to produce physically accurate training sets for ATR algorithms through synthetic images. Historically, the majority of the use of synthetic image generation (SIG) tools was to model the spectral phenomenology of target signatures because the sensor platforms were located either in space, as in the case with satellite based sensors, or at very high altitudes, in the case of aircraft based sensors. These high altitude platforms generally worked with ground sample distances (GSD) on the order of meters, or even kilometers, and produced images where manmade targets of interest were usually entirely contained within a single pixel. Therefore, no detailed spatial information about the target could be gathered in real world data collections and modeling fine spatial characteristics of those targets was not a priority. Now, with the advent of new technology, such as very high resolution airborne and space-based sensors, unmanned aerial vehicles (UAV) and forward looking optics mounted on low flying aircraft and helicopters, there is the capability to gather high spatial resolution target information. These new platforms are driving SIG tools to produce oblique,

high resolution images as opposed to the conventional nadir, low-resolution images. The requirement for high spatial resolution dictates that the scenes needed to be both spectrally and spatially correct.

A challenge with respect to ATR algorithms was dealing with the problem of appropriate background clutter. Detection algorithms generally look for small targets, but the modeler cannot focus solely on the accuracy of those few pixels that encompass the target; the entire synthetic image must be accurate. It is very important that a diverse background surrounds that target so that the algorithm will have sufficient variability in the synthetic image it is operating on to perform as it would on real data.

Size and positioning of targets with respect to the sensor and the scene were also important. Some target detection algorithms are both spectrally and spatially sensitive. This requires that not only the material properties of the target be accurate, but also it's orientation in the scene. Ensuring similar detection results between the truth and synthetic imagery requires that these key details be addressed.

The real world scene that the simulation is based on is located in a wooded area at RIT that the Center for Imaging Science has dubbed "MicroScene" (see Figure 1.2). This area has been extensively ground truthed. Detailed spectral and spatial measurements of the the area were taken to create the necessary database of information that was used in the synthetic image generation process.

Verification and validation are crucial parts of any modeling software package. The process ensures that the underlying assumptions about nature are correct and it gives the user of the synthetic imagery a measure of its quality. Confidence in the radiometric accuracy of the synthetic scene is especially important when the image will be used to train ATR algorithms. Obviously, poor training data will lead to incorrect classification and target recognition (garbage in equals garbage out).

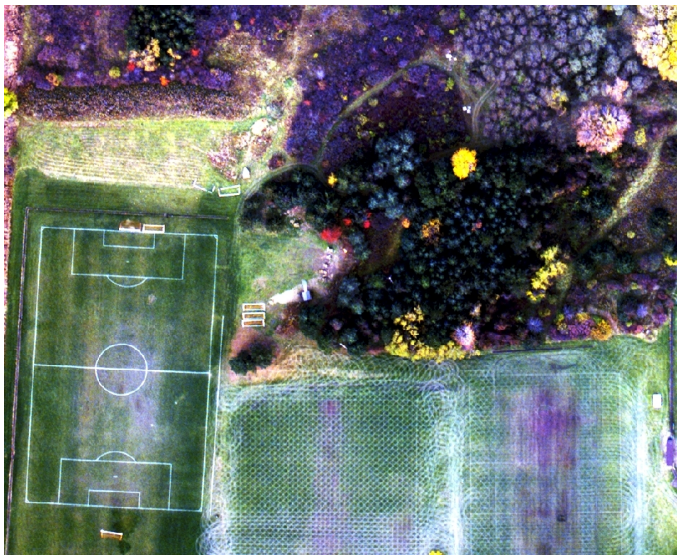


Figure 1.2: WASP image of MicroScene area at Rochester Institute of Technology taken in the fall.

Various methods were used to ensure the statistical and spatial integrity of the scenes that were generated in this project. As with any image comparison, the first step is to simply visually inspect the imagery for general similarities. This qualitative comparison method is used for both the image and spectral comparisons for achieving a baseline measure of the accuracy of the synthetic image. Informal comparisons like this are vital for making priority determinations about necessary improvements to the model from one development cycle to the next.

Once the initial comparisons display an acceptable level of accuracy then more numerical comparisons can be made to assist in the model's fine tuning. Since specific phenomenology for doing the concealed target detection had not been decided upon by the other universities associated with the Army MURI, four numerical metrics were chosen that the author felt best encompassed all of the areas that detection algorithms generally rely on to accomplish their objective. First, rank-order correlation (ROC)

is used to measure the contrast of significant targets of interest in the scene. Then, general image classification is accomplished with the Gaussian maximum likelihood (GML) classifier. Also, the spatial texture of the scene is measured using grey-level co-occurrence matrices (GLCM). Finally, the spatially and spectrally sensitive RX algorithm, is run on the data to address the goal of target detection. The data from the measurements is used to compare the results of running those same processes on the truth imagery of the scene. The differences and similarities provide an understanding of the success of the simulation effort.

The rest of this document will present an overview of some of the SIG tools that are available for this type of simulation. It will also describe the mathematical theory behind ROC, GML classification, the RX algorithm, and gray level co-occurrence matrix derived texture metrics. Then, an outline will be given of the setup for the MicroScene truth imagery collection and also the methods used to construct the synthetic model of that scene. The last portion describes the results of the analysis.

This work has been organized in various ways to assist readers who are both looking to accomplish a similar task and those who are looking to expand the use of the model. When the development of the MicroScene model is described, it is a "How-To" manual of sorts. This will hopefully assist any reader who is embarking on a similar line of research. Future users of the model who are looking to expand its utility should read this work with the understanding that the phenomenology that is validated is intentionally generic in an attempt to determine the model's ability to operate under the scrutiny of standard image classification and target detection algorithms. As algorithms are developed to exploit specific phenomenology, then that phenomenology may need to be included in the model before it will perform appropriately.

The results of the validation given the previous assumption are largely successful.

They are an encouraging first step forward in the ability of the DIRSIG simulation tool to supplant the need for actual image acquisition for training and developing detection and classification algorithms.

Chapter 2

Background

2.1 Synthetic Image Generation

Synthetic imagery is a critical component of Imaging Science for many reasons. It allows sensor designers the opportunity to create virtual versions of a sensor without many of the problems associated with creating costly physical versions. SIG tools also allow system users to determine the best way to utilize a particular sensor design. Those users can model prospective scenes and determine the combination of parameters, such as acquisition time, view angle, and weather conditions that maximizes the benefit of acquiring a real image. Another major benefit to computer modeling is the ability to do detailed error analysis. SIG tools allow a user total control over the process, so that physically impossible experiments, like completely removing the atmosphere or removing noise from the system, are possible. This allows designers to study exactly where the problems are in the system and determine how much each piece contributes to the overall system error (Schott, 1997, p. 363-364).

The major focus of this research will be for studying how beneficial current SIG tools

are in the development of hyperspectral ATR algorithms. The parameters that effect the capabilities of the algorithm, such as resolution, view angle, and spectral region, can be changed much more readily in a computer than having to create physical versions of the same scene. The algorithms can then be tested against a much more rigorous data set. In addition to ATR validation, SIG tools can be used to help train ATR algorithms. Most common ATR algorithms use statistical or neural net classifiers and therefore require a great deal of training data. Multiple scenes must be imaged for use as training data for ATR algorithms in order to increase the algorithm's understanding of the different conditions in which a target may be found. This is especially true when training hyperspectral algorithms. Duda et al. (2001, p. 84) explains how most ATR algorithms suffer from a lack of training data and that the problem is compounded when the dimensionality (e.g. number of spectral bands) of the problem is large. SIG can be a very useful tool for populating the training data when real data is not available. Although, according to Schott et al. (1998), it is generally not a good idea to use SIG exclusively for this purpose. They conclude that some real data should always be used to ensure that the SIG models do not introduce biases or artifacts that distort algorithm performance trends. The sensor designs, atmospheric conditions, resolutions, spectral regions, and targets can all be modified with a fraction of the cost that would be required when acquiring the information from real-world scenes. Schott et al. (1998) and Schott et al. (1999) give examples of the use of DIRSIG in ATR algorithm development.

Finally, the development of statistical and first-principles based computer simulations validates our understanding of the world around us. There is a certain amount of satisfaction that comes with the knowledge that the time spent researching and developing an understanding of nature has not been spent in vain. Doing this verification and validation of the DIRSIG model will hopefully accomplish that objective.

All of this requires that the SIG model be physically accurate at all levels of the scene that are of importance to the sensor being modeled or the algorithm being trained. The creation of a SIG scene is a very time consuming process because of all of the real-world information that must be acquired and cataloged before the simulation can be run. Also, all of the significant underlying phenomenology of the physical world must be understood in order to model it correctly.

There are many SIG tools currently available that are potential platforms for conducting this research. Most of these tools originated as a means to evaluate specific aspects of physical phenomena, such as thermal characteristics of objects or the effects of the human visual system. The ones that will be examined here are those that have evolved into major platforms that attempt to model a broad range of those phenomena. They are physical models, Irma, CAMEO-SIM, VISEO, and DIRSIG. Each of them has their strengths and weaknesses, but they have all increased the scientific communities understanding of how to capture the complex interactions between photons and their environment.

2.1.1 Physical Miniature Models

Before a complete study of SIG tools can be accomplished it is important to understand that SIG tools are not limited to computer simulation. Sometimes it is prudent to use miniature models of scenes to create synthetic imagery. Therefore prior to creating a synthetic image one must understand the tradeoffs between computer and miniature modeling so that the appropriate simulation method can be chosen. In the past, computer processing power limited synthetic image generation strictly to this type of modeling. When using physical miniatures, viewing angles are controlled by placing the scene on a rotatable and tiltable table. Illumination conditions are modeled with an

array of lights located at nearly every angle around the scene to simulate downwelled skylight. Sunlight is modeled by a high intensity light aimed directly at the scene from the appropriate angle. The sensors used are typically the actual sensors that will be used in the final design. The models are custom built by modelmakers and are only spatially limited by the modelmaker's abilities.

Miniature models are excellent for use in gathering reflective data in the 0.4 to 2.5 μm spectral region. Also, by their very nature they incorporate all of the complex interactions of the targets and their backgrounds instantly. The process is limited by a few factors, though. First, this is not generally a good technique for developing sensor designs since the sensor must be physically present. Another limitation is in the self-emissive regions of the spectrum. Thermal Infrared (IR) collections are enormously complex. It is practically impossible for miniature models to have similar emissivities, heat capacities, and active heating and cooling sources to real scenes. On top of that, the laboratory produces radiance that contributes to erroneous IR measurements. Francis et al. (1993) gives a detailed comparison of miniature modeling versus computer-generated models. The rest of this section will be devoted to computer-based solutions to synthetic image generation beginning with Irma.

2.1.2 Infrared Modeling and Analysis (Irma)

The Infrared Modeling and Analysis (Irma) tool was developed in 1980 by the Munitions Directorate of the Air Force Research Laboratory (AFRL/MNG). Its focus was to provide high-resolution infrared target and background simulations for smart weapons research and development. Originally, Irma was solely an IR tool, but has since gone through a series of upgrades to include both added phenomenology and increased user friendliness. The current version, Irma 5.0, includes modeling capabilities for three

primary signature channels. The first is passive sensor model for IR, Millimeter Wavelength (MMW), near IR (NIR), ultraviolet (UV) wavelengths. The last two channels are for modeling active Laser Radar (LADAR) and radar sensors. A common scene description file for each of the channels provides automatically registered images. Irma 5.0 includes utilities for editing scene facets and their properties, image viewers, target placement tools, and a Motif graphical user interface (GUI). A complete overview of this latest version was conducted in Wellfare et al. (2000).

Irma incorporates 3-D scenes developed in AutoCAD. The scenes are based on triangular facets with attached thermal, reflectance, scattering, and texturing properties. Separate thermal files can be input by the user to override the material properties for controlling difficult to model thermal conditions, such as engine operation or heat flow through pipes. Additionally, temporal files can be included into the simulation to control the frame-to-frame motion of targets, objects and the sensor in the scene.

Since the focus of this research does not include LADAR or radar studies, only the passive channels of Irma will be discussed here. Detailed descriptions of the capability to model LADAR and radar in Irma can also be found in Wellfare et al. (2000). The passive channels are modeled using three separate programs: ENVIRO, PASSIVE/PPASSIVE, and SSW. ENVIRO is the portion of Irma responsible for calculating the temperatures of the facets in the scene. It is based on thermodynamic first principles and is capable of computing one-dimensional heat transfer between objects as well as internal and external energy transfer within materials. PASSIVE and PPASSIVE use a z-buffer to render images based on Equation 2.1.

$$N = \tau N^s_{sun} + \tau N^s_{bkg} + \tau N^t + N^A \tag{2.1}$$

In that equation, N[1] represents the sensor reaching radiance, N^s_{sun} accounts for reflected solar photons, N^s_{bkg} accounts for reflected skylight and background illumination, N^A represents the upwelled radiance, and τ is the path transmission. The reflective properties of each facet are user-specified as a function of wavelength and are manifested as a combination of diffuse and specular reflectivity.

Equation 2.1 provides a lot of incite into the fidelity of the model. Irma does not provide a mechanism to explore the effects of reflections between objects in the scene. Also, the shape factor assumptions about the scene are generic and are applied uniformally throughout the scene. Lastly, the SSW package provides Irma with the capability to model complex sensor effects. Though polarimetric effects are not important to this research, the passive channels in Irma are coded so that modeling of both the polarized and unpolarized signatures of objects is possible.

Irma will probably not be effective for this research for a few reasons. The first is that there is no radiance contribution due to the surrounding background of targets because of Irma's z-buffer engine. Also, the current setup for modelling images allows for spectral images, but each one needs to be completed separately, creating a great deal of additional work to recreate the spectral data obtained from the sensors in this study. The next section will continue the overview of different SIG tools with a description of CAMEO-SIM.

2.1.3 Camouflage Electro-Optic Simulation System (CAMEO-SIM)

As opposed to Irma's z-buffering technique for creating 2-D radiance maps from 3-D scenes, CAMEO-SIM uses ray-tracing algorithms to provide true 3-D object interactions. CAMEO-SIM was developed in the United Kingdom as a physics-based, broad-

[1] L is often used as the variable that represents radiance. The author chose to use N here to maintain a similarity with the Irma references used for this section.

band (0.4 to 14μm) simulation tool with a main focus on studying camouflage, concealment, and deception. It provides scalable image resolutions which are defined by the user, thus allowing flexible trade-offs between the level of object detail, radiometric accuracy, and run-time. Also, the model contains a complete record of the material properties used by the renderer and the final image for detailed error analysis. A complete overview of CAMEO-SIM is given in Moorhead et al. (2001) and detailed results of validation experiments are described in Moorhead et al. (1999), and Mitchell et al. (2000); Mitcell et al. (2000, 2002).

CAMEO-SIM simulations begin with the construction of a 3-D scene in a CAD software package. Each object in the scene is modeled using textured facets. The texture values in the facets are linked to associated bi-directional reflectance, solar absorptivity, conductivity and density. Due to its true 3-D rendering capabilities, complex object interactions, such as shadows and background shape factor calculations can be done. Spatial resolution is only limited by the scene polygon count. Temporal properties of both objects and the sensor can also be added to the simulation as of CAMEO-SIM Version 2. Interestingly, the movement of the sensor through the scene does not have to be fixed ahead of time (although it can be). CAMEO-SIM provides an option for scene determined sensor paths based on interactive feedback from each generated frame of imagery.

The radiometry equations (2.2 and 2.3) utilized by CAMEO-SIM contain terms for wavelength and angular dependant thermal self-emission, atmospheric effects, and global illumination. Equation 2.2 solves for the radiance, at wavelength λ, leaving a point x, y in the scene in a direction (θ_0, ϕ_0). Equation 2.3 describes the sensor reaching

radiance for a sensor placed at coordinates x', y'.

$$\begin{aligned} N_0(\lambda, x, y, \theta_0, \phi_0) &= \epsilon(\lambda, x, y, \theta_0, \phi_0) N_{bb}[\lambda, T(x, y)] \\ &\quad + \int_{\cap} \rho_{bd}(\lambda, x, y, \theta_0, \phi_0, \theta_i, \phi_i) \\ &\quad \times N_i(\lambda, x, y, \theta_i, \phi_i) \sin(\theta_i) \, d\omega_i \end{aligned} \tag{2.2}$$

$$\begin{aligned} N_s(\lambda, x', y', x, y) &= \tau(\lambda, x', y', x, y) N_0(\lambda, x, y, \theta_0, \phi_0,) \\ &\quad + N_{path}(\lambda, x', y', x, y) \end{aligned} \tag{2.3}$$

The terms in these equations that solve for the self-emitted terms are characterized by ϵ, the emissivity, and N_{bb}, the blackbody radiance at temperature T (as described by the familiar Plank equation). Atmospheric effects are described by τ, the target to sensor transmission, and N_{path}, the upwelled path radiance. Finally, the global illumination, is shown as the integration of the incident radiance, N_i, around the hemisphere above the target multiplied by its bi-directional reflectance, ρ_{bd}, and angle of the incident radiance from the target normal, $\cos(\theta_i)$. Originally, the atmospheric and thermal terms were computed using MOSART and TERTEM. This limited the users to a selection of only 19 different atmospheres. A recent upgrade to CAMEO-SIM includes an interface to LOWTRAN and MODTRAN4 to provide atmospheric databases of spectral data for the required scenario (Filbee et al., 2002).

Prior to generating synthetic imagery using CAMEO-SIM, the user must determine the level of fidelity necessary to accomplish their goal. Output imagery can range from low fidelity imagery collected in real-time to processing time intensive, high fidelity imagery where all of the complex spectral and spatial qualities of the real world are recreated as closely as possible. Low fidelity modes decrease processing time by making a

series of simplifying assumptions including, simplified BDRF, no background-to-object or object-to-object interaction, and also restricting the number of facets in the scene. Use of these simplifying assumption can be very detrimental to the realism of a scene. Scenes useful for ATR algorithm training would have to be built using high fidelity modeling which includes BRDF effects, geometric occlusion of point and extended sources, and spectral integration of radiance to include the effects of the atmosphere.

Moorhead et al. (2001) provides example images of scenes created at low and high fidelity for comparison. From these examples it is clear that when attempting to model realistic camouflaged targets higher fidelity modeling is required and the low fidelity mode should only be used as a preview generator. One example shows the importance of both the Earth's thermal reflectance as well as albedo terms when modeling aircraft-to-sky contrast in the mid-wave IR (3 - 5μm). Another example demonstrates the unique ability of CAMEO-SIM to model active countermeasures. In that example, an aircraft releases active countermeasures in the form of chafe (used to confuse heat-seeking ordinance). The chafe provides an additional light source in the scene and its illumination can be clearly seen reflecting off of the underside of the aircraft. Before RIT's SIG tool is discussed, there is one more popular piece of modeling software to go over. It is Georgia Tech's VISEO.

2.1.4 Visual and Electro-Optical (VISEO) Detection Analysis System

The Georgia Tech Research Institute (GTRI) has developed an integrated suite of software for Visual and Electro-Optical (VISEO) detection analysis. The VISEO system is a comprehensive workstation-based tool for multi-spectral signature analysis, low observable (LO) design, and visualization of targets moving through real measured backgrounds. The program is designed to model the detectability of objects rendered

over real backgrounds while also simulating sensor and human visual system effects. VISEO is built upon a variety of smaller modules that perform the necessary radiance calculations, sensor effects and control view angles and object locations (Doll et al., 1997).

Essentially, scene generation is accomplished in two parts. The first is the development of the background terrain which can either be derived from measured data (Digital Feature Analysis Data (DFAD) and Digital Terrain Elevation Data (DTED) from the National Geospatial-Intelligence Agency) and satellite imagery or completely synthesized. The background is represented in grid format so that individual facets in the grid can be assigned appropriate material properties. While the majority of materials are assigned properties based on empirical data, several important ones, such as six varieties of soil, foliage, asphalt, concrete, water and snow are all based on first principles (Doll, 1997; Sheffer et al., 1996). The second part of the scene generation process is the incorporation of objects and targets on top of the background. These objects must also be created in a facetized format through the use of CAD software or the GTRI developed code MAX (Sheffer et al., 1996). Various levels of modeling fidelity can be used to create these objects, but in order to take advantage of all that the thermal model (described shortly) has to offer, then the modeler must also incorporate an internal nodal network of the object. These objects are placed in the scene using TARGLOC which provides the user with a friendly interface for pinpointing object locations and orientations as well as that of the sensor platform.

Various levels of detail can be used to assign radiance values to objects in the scene in VISEO. They range from simple blackbody calculations using the Planck equation, to the semi-empirical Irma model discussed earlier, to a first-principles approach in the form of the target signature model, GTSIG. GTSIG predicts hardbody emissive

temperature and blackbody radiance signatures and incorporates the primary thermal phenomenological effects such as thermal exchange between facets and the sky, aerodynamic heating, solar irradiation, convection and mass transfer (evaporation, condensation, icing, etc.) (Cathcart and Sheffer, 1991; Doll, 1997). Recently, GTSIG was pushed to produce high-resolution spectral image cubes into the reflective and midwave regions of the spectrum rather than merely broadband radiance images in the 8-12μm band. It was also expanded to include non-uniform sky radiance, such as a 3-D cloud model (Cathcart and Sheffer, 1991), and a flexible BRDF model.

The rendering process is accomplished under the GTVISIT submodel. GTVISIT begins by reading the input files to acquire the viewing geometry and then calculating the viewing transformation matrices needed for 3-D perspective projections onto the 2-D view screen. A z-buffer algorithm uses the range values of each element in the scene to determine which objects are hidden from the sensor's view and which will be used for determining the sensor reaching radiance at that particular pixel. After the z-buffering is finished the radiance values are attenuated according to the atmospheric parameters of the scene. The radiance map produced by this process is unaffected by senor MTF affects or noise. These things are incorporated through the GTSENSE submodel.

At this point, the top level VISEO software takes over. VISEO's camouflage editor routine allows the user to specify three-dimensional patterns of paints on the surface of the target vehicle. The patterns are described by a mathematical function so that an infinite number of possible patterns can be created. The number of different paints or coatings in the pattern is unlimited as well. Each of the paints is tied to material properties in the same way that was described above. Another submodel, GTV, is available that is designed to predict eye movements and detect targets in a manner that mimics as closely as possible the way the human visual system works. The GTV

submodel can be used to analyze the effectiveness of the camouflage created with the VISEO tool (Doll, 1997).

VISEO is also compatible with the Georgia Tech Synthetic Imaging Missile Simulation (GTSIMS) tool. The focus of this tool is not to provide greater capability to model camouflage, but to support applications, including analysis of aircraft susceptibility to IR missile threats, analysis of IR counter measures (IRCM) system effectiveness, and overall test and evaluation programs (Sheffer et al., 1996).

All of the SIG tools discussed to this point have unique capabilities that make each of them ideal for certain projects. Both CAMEO-SIM and VISEO could possibly be excellent options for this research if they were more readily available for use at RIT. Also, access to much of the documentation needed to fully understand the principles used to create finished products with these pieces of software is limited. Therefore, the final and most extensive overview of a SIG tool will be devoted to RIT's own DIRSIG software.

2.1.5 Digital Imaging and Remote Sensing Image Generation (DIRSIG)

The DIRSIG model is an integrated collection of independent first principles based submodels which work in conjunction to produce radiance field images with high radiometric fidelity. This modular design creates a high degree of flexibility and interchangeability within the model, as well as the capability to diagnose and improve the model by isolating and analyzing each submodel. DIRSIG has evolved over nearly two decades, from a Long-Wave Infrared (LWIR) SIG modeling tool originally developed for ATR algorithm development and sensor trade studies, to a high fidelity model aimed at producing hyperspectral images over the visible through long-wave infrared spectral range (Schott et al., 1998). There are five primary submodels of DIRSIG. They are

the Scene Geometry Submodel, Ray Tracer Submodel, Thermal Submodel, Radiometry Submodel, and Sensor Submodel.

As usual, image generation in DIRSIG begins by creating the geometric information of the scene. 3-D wireframe models are built in a CAD environment or can be purchased from commercial drawing providers. At RIT, object models are generally built using the Rhinoceros CAD software. Then, objects are gathered into scenes using a locally developed piece of software called, Bulldozer. The scenes created using those packages are then facetized. Facets are the most basic building block in the scene and, in their most basic form, each one representing a unique material type (See Figure 2.1). Each facet contains information about its location in the scene to include its vertices, surface normal, slope angle, and azimuthal angle. The material properties include temperature, self-generated power, specific heat, thermal conductivity, mass density, specularity, lobe width, visible emissivity, thermal emissivity, exposed area, thickness, optical description, emissivity, texture, and extinction coefficient (if transparent).

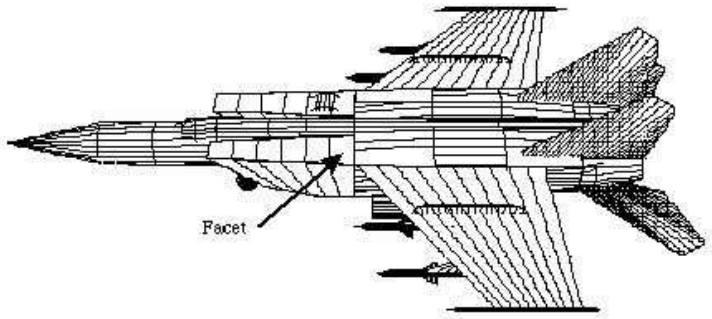


Figure 2.1: A wire-frame representation of a single DIRSIG object used to construct scenes (Brown, 2003a).

The Ray Tracing Submodel converts the three dimensional scene data into a two-dimensional image that conforms to the specifications of the sensor (linescanning, push-broom, framing, etc.) as well as the platform's motion through the scene. First, the

location, orientation, and characteristics (i.e. focal length, aperture diameter, etc.) of the sensor in the scene is established. Then, rays are cast from the sensor, through a pixel in the image plane, onto the scene as depicted in Figure 2.2.

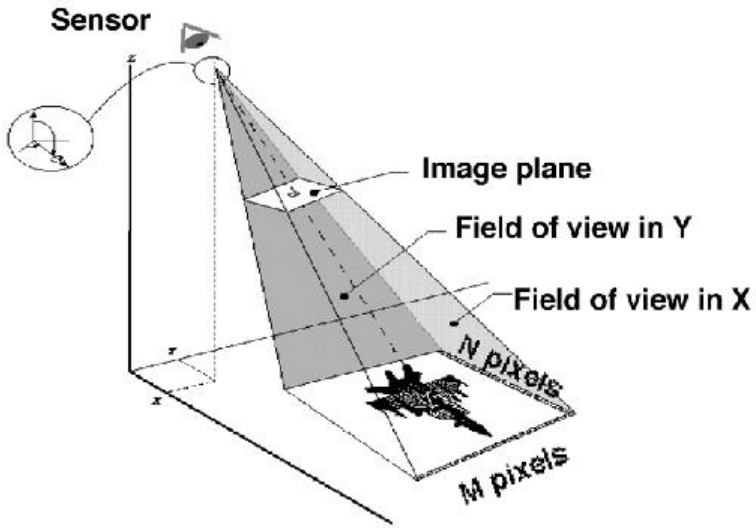


Figure 2.2: A ray tracer is utilized to determine which objects contribute to the radiance for a given pixel (Brown, 2003a).

The number of rays cast through a pixel can be increased to allow for a natural mixing of materials in the field-of-view of the pixel. In DIRSIG, the pixel is simply divided up into an NxN window of user-defined size. The divisions are treated as new pixels and the radiance values obtained by them are combined through weighted averaging to obtain a radiance value for the original pixel. This process is called oversampling. Without oversampling the entire radiance value for the pixel will be determined by the single material type encountered by the ray. To illustrate the importance of oversampling, consider the following example. First, assume that a virtual sensor platform has a one-meter resolution and that no oversampling is used. If a ray is cast through the center of one of that sensor's pixels and happens to intersect a lug nut on the tire

of an aircraft, then the entire pixel would contain the signature of the lug nut. The spectral characteristics of a one inch lug nut would obviously not account for the entire spectral makeup of the one meter pixel. When oversampling is used, the rubber tire, the wheel rim, the support frame, and maybe even the asphalt will all be included by being combined in a linear fashion to create a more accurate representation of what the sensor reaching radiance should actually be. Figure 2.3 shows a virtual conifer tree rendered both without oversampling and with 3x3 oversampling. There are obviously softer edges and more realistic detail in the image on the right. Unfortunately, the use of oversampling requires more rendering time.

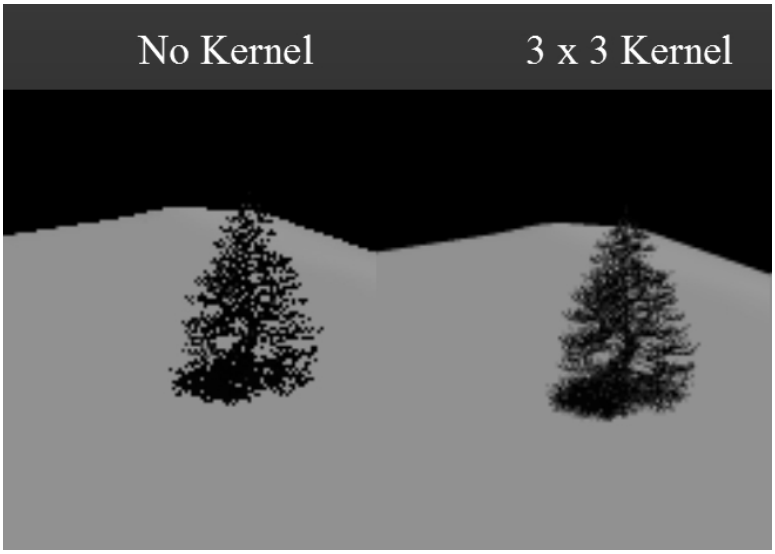


Figure 2.3: Example rendering of the effects of oversampling

Once the number of rays to be cast are determined, each one of them is processed in the same way. If the ray intersects a facet along its journey then the facet is checked to see if it is opaque or transparent. If it is transparent then the ray is attenuate by the transmissivity of the facet and the ray continues on until it strikes another facet. If the facet was opaque, then its emissivity is looked up in the database of the facet's

material properties. Reflectivity is calculated from the emissivity file. The specular and diffuse components of the material's reflectivity are in the DIRSIG material file.

The specular contribution is determined by casting two more rays, one to the sun and one in the specular direction to determine the radiance hitting the facet from the sun or from another background object. The radiance from these two sources is attenuated by the reflectivity of the facet and the angular dependent specular falloff values. When the BRDF model is used, the diffuse contribution is similarly calculated by casting multiple rays into the surrounding hemisphere of the facet to determine the downwelled radiance that it is receiving and the background shape factor. Once all this has been accomplished, the emissive, specular, and diffuse components are summed, multiplied by the atmospheric transmission, and added to the upwelled radiance. The result is the radiance reaching the sensor associated with that particular pixel.

The Thermal Submodel assigns temperatures to facets within a scene based on their material properties (i.e. heat capacity, thickness, density, solar absorption coefficient, etc.) and meteorlogical histories (i.e. wind speed, air temperature, dew point, temperature, solar insolation, etc.). These temperatures are predicted by a forward chaining differential model called THERM, which was written by DCS Corporation as part of the Air Force Infrared Simulated Image Model (AIRSIM) (DCS Corporation, 1991). If the thermal model is being used, then when a ray intersects a facet the sun/shadow history of that facet is computed in addition to its reflectance and transmission. The sun/shadow history is computed by casting a ray from the facet to the location of the sun at set time intervals in the facets history. If the facet is in direct sunlight, or illuminated through a transmissive object, then that information will be used to calculate the time dependent temperature of the facet based on its material properties.

One drawback of THERM is its inability to model lateral conduction between ad-

jacent facets. To alleviate this, DIRSIG allows the user to override THERM and input data from some other thermal modeling software package, such as MuSES. MuSES is currently being upgraded to enhance its ability to include environmental and background radiation effects from extensive scenes and may become a potential substitute for THERM in DIRSIG (Schott et al., 2001).

The Radiometry Submodel is the mathematical engine behind the Ray Tracing Submodel. This module contains the underlying mathematics that model the physical processes as well as the Air Force's MODTRAN atmospheric propagation model. MODTRAN is used to determine exoatmospheric irradiance, emitted and scattered radiances (upwelled and downwelled) and path transmission predictions. The calculations are computed spectrally at a maximum spectral resolution of 2 wavenumbers. The results are based on a standard or user-supplied atmospheric profile, latitude, longitude, elevation, time-of-day, and day-of-year (Berk et al., 1999). This submodel utilizes bidirectional reflectance data, accounts for specular and diffuse background contributions, and features path length dependent extinction and emission for transmissive bodies (plumes, clouds, etc.) which may be present in any target, background or solar path. The radiometry model simply processes the rays generated by the ray tracer to compute a spectral radiance reaching the sensor for a given pixel (Brown, 2003a).

It is important to understand the different types of photons that may be gathered by a sensor before tackling the underlying mathematics. Figure 2.4 shows the paths that a photon may take to reach the sensor. These paths can be collected into two main groups: solar and self-emitted. This grouping is useful when considering which wavelengths will be modeled in the scene. Solar photons have more impact in the 0.3 to 2.5 μm region and self-emitted photons are dominant in the 8 to 14 μm region.

The mathematical physical model at the heart of the Radiometry Submodel is called

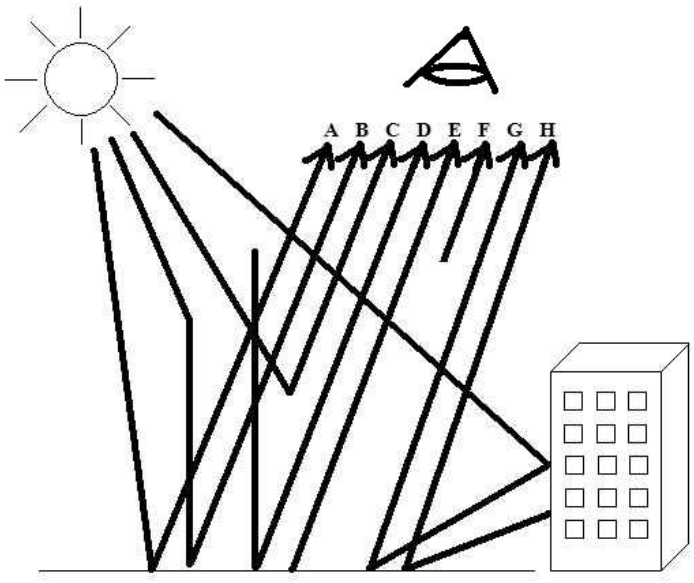


Figure 2.4: Solar and Self-Emitted Photon Paths (Schott, 1997, p. 114)

The Big Equation by Schott (1997, p. 113). It is presented here in Equation 2.4. Table 2.1 gives a definition of the variables in the big equation. It also sheds some light on the photon paths that the variables correspond to.

$$L_\lambda = \left\{ E'_{s\lambda} \cos\sigma' \tau_{1\lambda} \frac{r(\lambda)}{\pi} + \epsilon(\lambda) L_{T\lambda} + F\left[E_{ds\lambda} + E_{d\epsilon\lambda}\right] \frac{r_d(\lambda)}{\pi} + (1-F)\left[L_{bs\lambda} + L_{b\epsilon\lambda}\right] r_d(\lambda) \right\} \tau_2(\lambda) + L_{us\lambda} + L_{u\epsilon\lambda} \quad (2.4)$$

Even though facets are assigned individual material identities, the variability within that material may still need to be accounted for. For example, a grassy field will have different spectral characteristics depending on which spatial location of the field one is looking at. The differences are due to mixtures of materials (dirt, moisture, underlying

Variable	Meaning
L_λ	Wavelength dependent total sensor reaching radiance
$E'_{s\lambda}$	Exoatmospheric irradiance - A photons (after being reflected)
σ'	Angle from target to sun
$L_{T\lambda}$	Self-emitted radiance - D photons
τ_1	Transmission from sun to target
τ_2	Transmission from target to sensor
F	Shape factor of the background
r	Reflectance
ϵ	Emissivity
$L_{ds\lambda}$	Downwelled solar radiance - B photons
$L_{d\epsilon\lambda}$	Downwelled atmospheric self-emitted radiance - E photons
$L_{bs\lambda}$	Background solar reflected radiance - G photons
$L_{b\epsilon\lambda}$	Background self-emitted radiance - H photons
$L_{us\lambda}$	Upwelled solar reflected radiance - C photons
$L_{u\epsilon\lambda}$	Upwelled self-emitted radiance - F photons

Table 2.1: Summary of variable meanings and corresponding photon paths

rocks, etc.) and the health of the grass. Therefore, DIRSIG is capable of using a large database of reflectance curves for any given material. Each material class can be assigned a texture image that represents that spatial variation of reflectance in one specific spectral region. When a ray is cast onto a facet, a mapping mechanism identifies the location on the facet that corresponds to the same pixel location in the texture image. The value of that pixel is used to select a reflectance curve from the material database. That is accomplished by calculating the z-score, of that pixel within the texture map. Then, a reflectance curve is selected from the material database that has the same z-score (Scanlan, 2003).

Scanlan (2003) worked to evaluate the single-bandpass (SBP) z-score approach to reflectance curve selection against a variety of methods. The problem with the SBP approach is exactly like it sounds. It only uses a single band of data to determine the reflective curve to use across all bands of a material. In his thesis, a multiple-

bandpass (MBP) z-score technique was implemented into DIRSIG and a validation was accomplished which showed that the MBP technique proved to be much more robust than the SBP. In another model implemented in DIRSIG by Scanlan, texture is created by using spatial and spectral statistics determined from endmember fractional abundance maps using an automated endmember selection algorithm. The process behind this approach begins by obtaining a hyperspectral image of the scene being modeled and extracting the principal endmembers from it. Next, fractional abundance maps of each of the endmembers are constructed. These fraction maps are then assigned to the background plane of the synthetic scene. When a ray in DIRSIG encounters the background it will compute a linear combination of the endmember's spectra weighted by the values in the fraction maps and assign the result as the spectrum of that target.

Unfortunately, the current DIRSIG model only allows texture mapping onto the background of the synthetic scene. This is because all maps default to a vertical projection onto the mapped facets (Brown and Schott, 2000). Attempting to assign maps to facets who's normals are not vertical will result in a deformation of the map and an incorrect result. Currently, work is being done that will alleviate this problem and is projected to be available as of DIRSIG release 4.0. The goal is to modify DIRSIG so that textures are applied at the object level. To begin, each object is "unfolded" similar to that shown in Figure 2.5. Textures of the object are applied facet-by-facet and then those textures are "folded" around the object.

The proper orientation and scaling of the texture image is accomplished through interpolation. Essentially, the desired 3-D object texture is applied through a 2-D image with horizontal and vertical coordinates, u and v, as shown in Figure 2.6. This 2-D coordinate system is related to the 3-D coordinates of the object through $u = f_u(x, y, z)$ and $v = f_v(x, y, z)$. The function f represents a linear transformation that simply scales

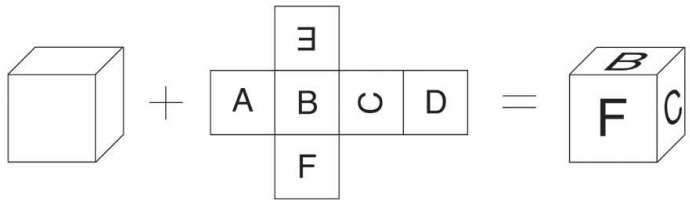


Figure 2.5: Simple example of object texture projection

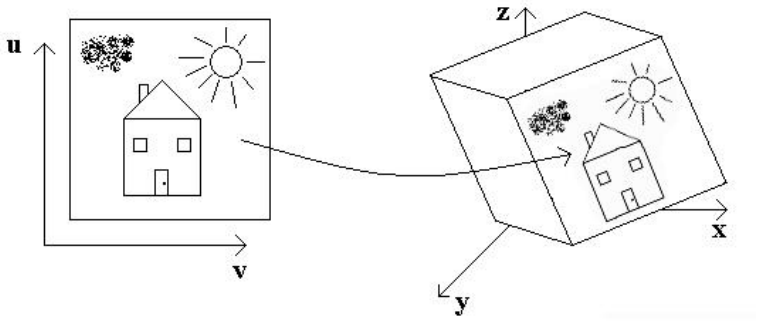


Figure 2.6: Projecting 2-D texture onto 3-D coordinates

the texture image across the facet. When the ray tracer casts a ray that hits the object facet, instead of looking for a material spectrum, it first computes the relative location of impact on the facet and then uses that information to select a pixel at the same relative location in the texture map. The pixel selected from the texture map will dictate the material spectrum selected for that ray.

The use of the object textures was expected to be available for use in this research, but unfortunately was not. Many of the virtual objects used in the creation of the MicroScene model were developed with texture images and embedded u, v coordinates. They will be ideal for testing the object texturing capability when it becomes available in DIRSIG 4. Figure 2.7 shows an example of the texturing methods used in a beta version of DIRSIG 4. The best example of the new texturing method can be seen in

the resolution chart model. Projecting the resolution bars onto this object in DIRSIG 3 would have produced inaccurate results because DIRSIG 3 uses a vertical projection method for doing object texturing. This method was originally created with terrain mapping in mind, since terrain plates are generally horizontal surfaces. The method is inadequate for nearly vertical objects such as the resolution chart. Also of interest in Figure 2.7 are the shingles of the shed and the wood ramp. These two areas have also have been modeled with the new texture wrapping technique.

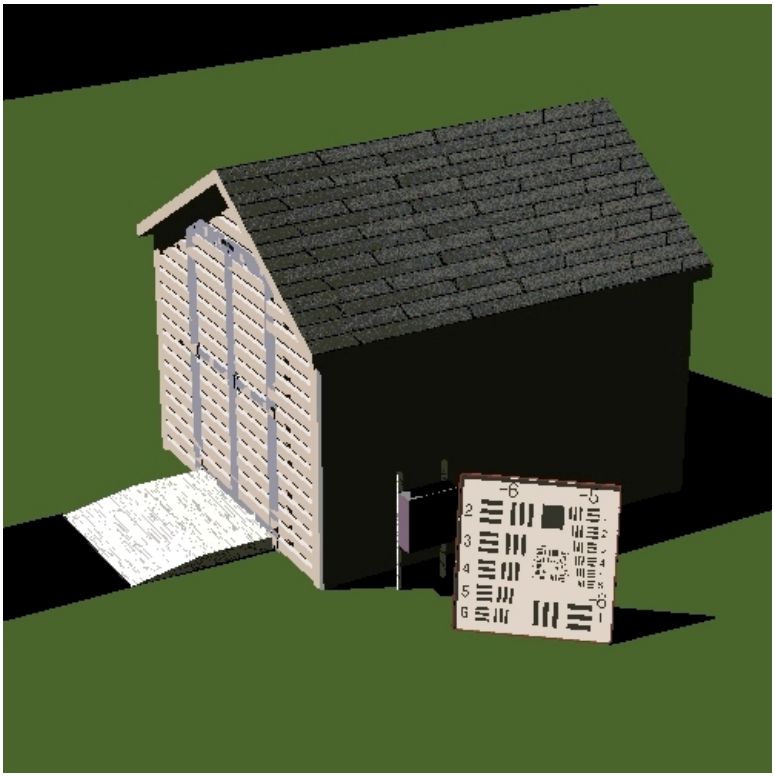


Figure 2.7: An example of texturing in a beta version of DIRSIG 4.

Another problem with facet-based simulation models is abrupt material transitions. If a scene contains a road with a gravel shoulder, a generic facet-based simulation would produce a very high contrast edge between the two materials. Of course, in the real

world, some gravel would fall on the road and create a transition region which would blur the contrast between the two regions. Material maps in DIRSIG allow the user to "paint" regions on top of a facet to spatially assign it more than one material property. Then, the transitional region in between two materials can be controlled to take care of the unrealistically high frequency transitions inherent in facetized environments. Brown and Schott (2000) present a more specific account of the various maps available in DIRSIG. An added benefit of material maps is that fewer facets are required so long as elevation changes are not a critical component of the material transition.

Variability within a material can also be enhanced through the use of bump mapping. Bump mapping uses gray-scale images that control the amount of deflection that should be added to a surface normal. The magnitude of the deflection is calculated from the gradient from one pixel to the next in the bump map image. Bump maps add to the realism of the scene through the appearance of surface roughness.

Finally, the Sensor Submodel converts the entrance aperture radiance to system output radiance through the system responsivity, which is modeled with Equation 2.5 (Schott et al., 1998). The Sensor Submodel does not have the capability to include noise or modulation transfer function (MTF) effects, but those should also be available in DIRSIG 4.0. They can be included in the final synthetic imagery, but they must be added in post-processing.

$$L_i = \int L_\lambda R_i(\lambda) d\lambda \tag{2.5}$$

Since its inception, DIRSIG's development has been focused on creating precise radiometrically accurate images through the use of first principles equations in conjunction with empirically measured, material specific emissivity databases. A great deal of research has been conducted on its accuracy and the software tools, documenta-

tion, developers, and users are all located at RIT. All of this makes DIRSIG the logical choice for modeling the MicroScene area. Also at RIT are the actual sensors that were used in the collection of the truth imagery. These sensor packages will be discussed in the next section.

2.2 Sensors

Truth imagery of MicroScene was obtained with two imaging instruments, the Multispectral Imaging Spectrometer Instrument (MISI) and the Wildfire Airborne Sensor Program (WASP). These two platforms were the logical choice for collecting the imagery because they are both designed, owned and operated at RIT. These sensors were hoisted into an elevated, oblique-viewing position via a scissor cart in order to mimic the view angle representative of forward-looking optical instruments.

2.2.1 Multispectral Imaging Spectrometer Instrument (MISI)

MISI is an airborne, line-scanning instrument with a 6" rotating mirror coupled with a f/3.3 Cassegrainian telescope. The instrument contains many sensors for broadband and multispectral imaging. MISI's broadband capability measures the visible, SWIR, MWIR, and LWIR regions of the spectrum. Two separate 36-channel spectrometers cover the electromagnetic spectrum from $0.44\mu m$ to $1.02\mu m$ at $.01\mu m$ increments. The ground instantaneous field-of-view (GIFOV) for the broadband visible is roughly 0.3 m and 1.0 m for the spectrometers, at an altitude of .3 km. The system has been used at RIT for high-altitude aircraft and satellite sensor performance evaluation, data collection for algorithm development, and as a survey instrument for demonstrating proof-of-concept studies in areas ranging from, water quality assessment to energy conservation (CIS, 2003).

2.2.1.1 MISI Noise Characteristics

Analysis has been done at RIT to determine the operating characteristics of the MISI sensor. The gain, bias and full-width half-max (FWHM) factors of each of the bands of the visible spectrometers has been described in previous laboratory experiments. The gains are found by pointing MISI's field-of-view (FOV) into an integrating sphere and then using a monochrometer to illuminate the sphere. Since the wavelength and intensity of the light from the monochrometer is known, the gain factor of each spectral band can be backed out of the digital counts obtained during the measurement. The bias factor is derived from the actual MISI imagery obtained during a collection. When obtaining images, the instrument is configured to gather 40-50 lines of data while the shutter is closed so that a dark region shows up at the beginning and end of the imagery. Because no outside signal reaches the detector in this area, the resulting digital count (DC) values are only representative of the bias level of the system and noise. If the noise is assumed to be additive and have a zero-mean value, then a simple averaging of all of the values in this area will reduce to the value of the bias. The gain and bias factors can be applied to the images generated by MISI to make them radiometrically calibrated by subtracting the average bias of the dark region and multiplying by the gain on a band-by-band basis.

The dark region of the image also can be used to describe how the noise associated with MISI is correlated from band-to-band. The correlation matrix of the dark region of one of a MISI image is shown in Figure 2.8, where the first band correlation starts in the upper-left hand corner of the figure. This image is also instructive about how the detector boards on MISI are arranged. For instance, boards 1 and 2 each contain a 16 channel spectrometer covering from about 400 to 700 nm at 20 nm intervals. One board is offset from the other by 10 nm so that 36 bands of data are obtained at 10 nm

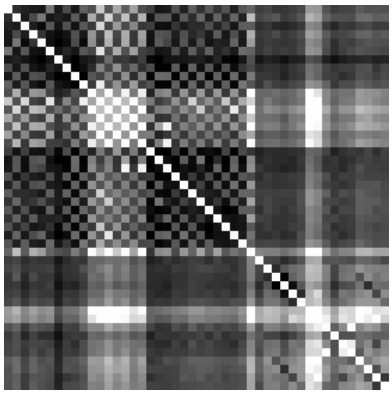


Figure 2.8: MISI Noise Correlation Matrix

spectral resolution. This interleaving is evident in the figure as the checkerboard-like pattern seen in the first 30 bands. The data will be utilized in Section 3.2.7 to build synthetic correlated noise cubes for inclusion into the DIRSIG rendered imagery.

2.2.1.2 MISI Point Spread Function

The optics and detector of the MISI instrument both create distortions in its image products. This distortion manifests itself as a "spreading", or blurring, of light from one pixel into adjacent pixels. If it is assumed that this light came from a single point source, then the distortion is referred to as the Point Spread Function (PSF). If it is also assumed that the PSF acts the same over all spatial locations in the scene (i.e. that it is linear, shift-invariant) then the PSF acts on the image as represented in Equation 2.6. In that equation, $*$ represents the convolution operation.

$$Image_{out} = Image_{in} * PSF \tag{2.6}$$

PSFs are difficult to measure experimentally. For this reason it is convenient to think in terms of a Line Spread Function (LSF). Assuming the PSF is rotationally

symmetric, the LSF is a 1-D slice through the 2-D PSF. This is easier to measure because one merely needs a knife edge to form a step function exposure. To further explain, assume that a scene is constructed such that one half of the scene is some uniform radiance and the other half is completely dark. Ideally, in a digital system, only the pixels on the radiant half would record any photons and the other pixels would remain at a zero DC level. Unfortunately, a real imaging system blurs the contrast at the boundary. Pixels just on the dark side of the edge, that should be completely dark, obtain some amount of light from the radiant half. Pixels just along the edge, but located on the radiant side, will not retain all of the photons that would have been imaged in the ideal case. The photons are spread between the pixels all along the edge. This phenomenon can be described as an Edge Spread Function (ESF). Mathematically, the LSF, shown in Equation 2.7 in the x direction, is simply the PSF integrated with respect to y.

$$LSF(x) = \int_{-\infty}^{\infty} PSF(x, y)dy \tag{2.7}$$

Furthermore, the ESF is given by Equation 2.8.

$$ESF(x) = \int_{-\infty}^{x} LSF(x)dx \tag{2.8}$$

After differentiating both sides of Equation 2.8 and rearranging, the result is Equation 2.9.

$$LSF(x) = \frac{d}{dx} ESF(x)dx \tag{2.9}$$

This means that the slope of the plot of irradiance vs. distance in Figure 2.9 is equal to the line spread function at that point (Schott, 1997, p. 320).

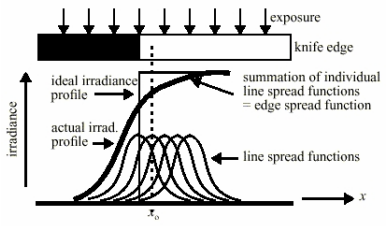


Figure 2.9: Illustration of ESF and LSF

Practically, the PSF can be approximated in real imagery by locating a region in the image where a bright, constant region transitions to a dark, constant region at an edge, such as the boundary between a building and the sky. In the case of a line scanner, such as MISI, the best approximation is to generate one LSF in the vertical direction and another in the horizontal. Then, interpolate a 2-D gaussian from those two LSFs.

2.2.2 Wildfire Airborne Sensor Program (WASP)

The other instrument being used in this study is the WASP system. WASP is really a compilation of four off-the-shelf framing array camera systems that provide a broadband imaging capability in the VNIR ($0.4 - 0.9\mu m$), SWIR ($0.9 - 1.7\mu m$), MWIR ($3.0 - 5.0\mu m$), and LWIR ($8.0 - 9.2\mu m$) regions of the spectrum. The VNIR 55mm focal length digital camera provides a 16 megapixel image at 12-bit color quantization. This camera is generally used for mapping purposes while the SWIR, MWIR and LWIR cameras take in information desired for the system's primary purpose of detecting wildfires (McKeown, 2003). Of the four cameras mentioned, the high-resolution mapping camera was used to collect the data necessary for this validation effort.

2.3 Verification and Validation

This research project is an effort to understand how well DIRSIG phenomenologically simulates the real world at high spatial resolutions, oblique view angles and different times of day. One's first inclination to conducting such an experiment would be to image a scene and then build an exact virtual replica of that scene for comparison on a pixel-by-pixel basis. If the focus of this research was a design effort where exact spatial dimensions were of the utmost importance, then that would probably be a correct assumption. Unfortunately, that type of comparison is extremely sensitive to geometric changes and alignment of the two images. It is a very difficult task to exactly match a synthetic image to the real world not only geometrically, but also for the same meteorological conditions (Mitcell et al., 2002). Imagine modeling the exact location and size of every leaf on a tree or blade of grass on the ground. As mentioned earlier, the goal of this research is to provide ATR algorithms with the data necessary to train them to detect concealed targets. Obviously, if the geometric layout of the target and its surroundings were known exactly, then there wouldn't be much need for the target recognition algorithm to begin with. Therefore, another method of determining the accuracy of the synthetic model is required so that exact spatial registration is not necessary.

One way to do this is to select various control points from both the simulated imagery and the corresponding truth imagery and then compare the relative brightness of those points. This process is known as rank-order correlation (ROC). Another way is to run classification and target detection algorithms on both types of images and compare the results for similarities. There are many different types of algorithms to choose from in the literature. The standard classification algorithm is the Gaussian Maximum Likelihood algorithm (Richards, 1999), which makes its determination based

solely on the spectral signatures of each pixel. This algorithm is already included as an ENVI tool which facilitates its implementation. Gray-Level Co-Occurrence Matrices are also a means of classifying an images except that they work based on the spatial qualities of the imagery. Manolakis (2002), includes an overview of the basics of target detection. Also in that paper, various geometric and statistical algorithms are compared as well as their usage with regard to full and sub-pixel targets of interest. The most interesting algorithm with respect to this research in that paper is the RX algorithm. It is a statistics-based algorithm that uses both spatial and spectral information about the target in the detection process.

These analysis tools provide a measure of the generic accuracy of the model. The goal of this research is not to determine which features will best discriminate targets from their backgrounds. That job is left for one of the other universities participating in the Army MURI (see Figure 1.1). The goal is to create synthetic imagery that performs similarly to the truth imagery when these algorithms are applied to them. Success is not dependent on accurate classification or detection, but rather similar classification and detection between the data sets. The next few sections will provide a more in depth overview of ROC, GML, GLCM and the RX algorithm.

2.3.1 Rank-Order Correlation (ROC)

ROC is a statistical measure of similarity between data sets. In a validation study, such as those that have been done on DIRSIG images in the past, Kraska (1996); White (1996), and Joseph (1998), the data points that populate the sets are carefully selected places of interest in the scene. The goal of applying ROC to the points from the two images is to obtain a measure of the relative contrast of the data points. Each data point in a particular image is assigned a rank-order according to its relative brightness,

or digital count. This is done independently in both the real and synthetic images. Then the two data sets are compared using Equation 2.10 (Mason et al., 1994) , which produces what is called the Spearman Rank-Order Correlation Coefficient, ρ.

$$\rho = 1 - \frac{6\Sigma(R_i - R_i^{'})^2}{n^3 - n} \tag{2.10}$$

In that equation, R_i is the rank of the i^{th} data point in the truth image, $R_i^{'}$ is the rank of the i^{th} data point in the synthetic image, and n is the total number of samples. If every point in both scenes is ranked in exactly the same way, then the coefficient will result in a value of one. Values less than one show that there is an inconsistency in the data points. This is also indicative of a problem within the model and that is the basis for the use of ROC in validation efforts.

The key to extracting the most information from ROC relies on an understanding of the process and a careful selection of data points. Remember, ROC is solely a measure of ranked contrast similarity between selected points from the two scenes. For this reason it does not require that the data points be registered. One thing that ROC does not do is provide any indication of gain or bias differences because variations in these parameters will not change the ranking of the data points [2]. The data points that are selected to be rank-ordered should be easily distinguishable in both the real and synthetic imagery and be chosen so that no mixing, or at least the same level of mixing, of materials occurs at the location.

[2]Root Mean Squared (RMS) error is generally used as a complementary measure of similarity alongside ROC because it is inherently sensitive to gain and bias changes between the images. Also, its results are indicative of overall accuracy as opposed to the accuracy of individual objects. RMS error requires that the two images be exactly spatially registered and is therefore unviable for this research.

2.3.2 Gaussian Maximum Likelihood Classification

Detection algorithms generally look for small targets, but the modeler cannot focus solely on the accuracy of those few pixels that encompass the target; the entire synthetic image must be accurate. It is very important that a diverse background surrounds that target so that the algorithm will have sufficient variability to perform as it would on real data. Gaussian maximum likelihood (GML) classification is a leading method in the remote sensing community for accomplishing this task because of its robustness and simplicity.

It works by examining the spectra of every pixel in an image and estimating the probability of that pixel belonging to one of i classes. The covariance, $\mathbf{S}_i$, and the mean, $\mathbf{M}_i$, of each class are provided to the algorithm as a priori knowledge. Then, the spectral column vector, which has l bands, of each image pixel, $\mathbf{X}$, is assigned a probability, $p(\mathbf{X}|i)$ of belonging to each class. The probability of $\mathbf{X}$ given i is shown in Equation 2.11 on the assumption that the ground truth data of class i will form a Gaussian distribution.

$$p(\mathbf{X}|i) = \frac{1}{(2\pi)^{\frac{l}{2}}|\mathbf{S}_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(\mathbf{X}-\mathbf{M}_i)^T \mathbf{S}_i^{-1}(\mathbf{X}-\mathbf{M}_i)} \tag{2.11}$$

The final likelihood value used to assign the pixel to a given class is shown in Equation 2.12. This equation weights the result of Equation 2.11 by $p(i)$, which is the probability of class i occurring in the image, and $p(\mathbf{X})$, which is the probability of $\mathbf{X}$ occurring in the image. The pixel is assigned to the class that generates the largest value of $p(i|\mathbf{X})$.

$$p(i|\mathbf{X}) = \frac{p(i)p(\mathbf{X}|i)}{p(\mathbf{X})}) \tag{2.12}$$

ENVI has a built-in version of the GML classifier based on Richards (1999). This version includes the capability to leave pixels whose probability of belonging to any class is low as unclassified. A user-defined threshold value determines level of confidence necessary to classify each pixel in the image. GML focuses on the spectral properties of the image, but completely disregards its spatial information. The next section will explore the idea of extracting spatial information through GLCM derived texture metrics.

2.3.3 Gray-Level Co-Occurrence Matrices (GLCM)

The focus of this research is to produce high-resolution synthetic imagery that mimics truth imagery, not only spectrally, but also spatially. How does one evaluate that success in a mathematically provable manner? The answer to this question, for the purpose of this research, is a combination of a qualitative review by human observation as well as the quantitative gray-level co-occurrence matrix (GLCM) derived texture metrics. GLCM can be used to extract several statistical spatial and angular relationships between neighboring gray-level values in an image. It was originally described in Haralick et al. (1973) for extracting textural information for image classification purposes. The driving factor behind Haralick's research was the ambiguous descriptions of image textures that had plagued analysts. He set about to categorize and segment various image texture types. The reason being, that the information contained in the textural properties of images were being overlooked and could be included in the classification process as additional features for enhancing data separability.

The GLCM is a second-order statistical texture measure. Walker and McManamey (1993), Baraldi and Parmiggiani (1995), and Doll and Home (2001) describe the importance of this when using statistical metrics to differentiate targets from background.

First-order statistics are only those statistics that can be calculated by considering the value of a single pixel at a time or by means of the image histogram. Values such as standard deviation and mean are considered first-order statistics. The first-order statistics of a particular target may not distinguish it mathematically from the clutter around it even though the shape and contours of that object make it easily distinguishable to the human observer. In fact, it is impossible for first-order statistics alone to differentiate between non-uniform, spatially distributed targets and non-uniform backgrounds (Walker and McManamey, 1993). Second-order statistics are those that simultaneously consider the values of two or more pixels which are spatially related to each other in some particular way. Second-order statistics can take greater advantage of these kinds of distinguishing features when mean and standard deviation do not provide the data separability required. Herein lies the importance of the GLCM metric for this research.

The GLCM technique finds the relationship of co-occurrence between two gray levels i and j at a given orientation α and distance d, for all possible occurring pairs in an image window of user defined size. To further explain, let m denote the number of gray levels in the window. The GLCM $P_{d,\alpha}(i,j)$ is an (m x m) matrix, where the entry indexed by i and j denotes the number of pairs of pixels separated by a distance d at and angle α which have gray levels i and j. $p_{d,\alpha}(i,j)$ is the normalized version of $P_{d,\alpha}(i,j)$ and is equivalent to $p_{d,\alpha}(i,j) = \frac{P_{d,\alpha}(i,j)}{m}$. Due to the constraints of discrete pixel windows, the orientations are usually 0°, for horizontal, 45°, for positive diagonal, 90°, for vertical, and 135°, for negative diagonal gray level relationships. When directional information is not required, or an additional feature vector is desired, isotropic versions of the directional GLCMs can be created through the use of Equation 2.13 (Walker et al., 1997).

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Figure 2.10: 4 x 4 image window with four gray tones

	0	1	2	3
0	(0,0)	(0,1)	(0,2)	(0,3)
1	(1,0)	(1,1)	(1,2)	(1,3)
2	(2,0)	(2,1)	(2,2)	(2,3)
3	(3,0)	(3,1)	(3,2)	(3,3)

Figure 2.11: General form of a GLCM for gray levels 0 - 3

$$p_d(i,j) = \frac{1}{2\pi}\int_0^{2\pi} p_{d,\alpha}(i,j)d\theta \tag{2.13}$$

Consider an example shown in Figure 2.10 through Figure 2.15. Imagine that Figure 2.10 represents the gray-level values of a subregion in some image. Figure 2.11 shows the GLCM layout where (i,j) stands for the number of times gray tones i and j have been neighbors as defined by the d and α. The integer number in each cell is determined by counting the number of pairs of pixels in Figure 2.10 such that the first pixel has gray level i and the second has gray level j. Two interesting properties occur as a result of this process. First, notice that the diagonal always contains even values because two adjacent pixels with the same gray-level value will increment the count both forwards and backwards. Second, and more important from a mathematical point-of-view, the GLCM matrices are always symmetrical. Figures 2.12 through 2.15 show four resulting co-occurrence matrices at the four orientations and $d = 1$.

The assumption being made in all of this is that texture of the area in the image under the window can be adequately described by the gray level relationships of the pixels in that window. By themselves, these matrices do not provide the desired tex-

$$P_{1,0^\circ} = \begin{pmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Figure 2.12: Horizontal GLCM

$$P_{1,90^\circ} = \begin{pmatrix} 6 & 0 & 2 & 0 \\ 0 & 4 & 2 & 0 \\ 2 & 2 & 2 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

Figure 2.13: Vertical GLCM

$$P_{1,135^\circ} = \begin{pmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{pmatrix}$$

Figure 2.14: Negative Diagonal GLCM

$$P_{1,45^\circ} = \begin{pmatrix} 4 & 1 & 0 & 0 \\ 1 & 2 & 2 & 0 \\ 0 & 2 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

Figure 2.15: Positive Diagonal GLCM

tural information. They can be considered a means to an end, so-to-speak. Haralick et al. (1973) described 14 measures of textural features that can be extracted from the GLCMs through various equations. The values solved for in these equations relate to specific textural characteristics of the image such as homogeneity, contrast, the presence of organized structure, complexity, and the nature of gray level transitions in the image. Unfortunately, it can be difficult to identify how each of the feature equations relates to a specific form of texture. Generally, it is not necessary to do so, as each of the features can be directly used as an additional spatial discriminating metric for image classification algorithms so long as when they are used together they enhance the separability of the data.

Many of Haralick's original fourteen features contain redundant information or are linearly dependant. Therefore, much work has been done to determine which features provide the most information with regards to data separability. Baraldi and Parmiggiani (1995) narrow Haralick's original fourteen metrics down to six. The six statistical parameters are *energy*, *contrast*, *variance*, *correlation*, *entropy* and *inverse difference moment*, which is also known as *homogeneity*. In that paper, *energy* and *contrast* are considered to be the most efficient at discriminating different textural patterns. Unfortunately, ENVI does not have the capability to compute *energy*, so *homogeneity*, *contrast*, and *entropy* will be used in this research. The decision to use those three arose from a conversation with the author of Scanlan (2003) who thesis was on using GLCM metrics to evaluate background textures in simulated imagery. Equations 2.14 through 2.16 show the equations used to derive the scalar values of those three features from the GLCMs.

$$Homogeneity = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} \frac{p(i,j)}{1+(i-j)^2} \tag{2.14}$$

$$Contrast = \sum_{i=0}^{m-1} \sum_{j=0}^{m-1} (i-j)^2 \cdot p(i,j) \tag{2.15}$$

$$Entropy = -\sum_{i=0}^{m-1} \sum_{j=0}^{m-1} p(i,j) \cdot log(p(i,j)) \tag{2.16}$$

Baraldi and Parmiggiani (1995) also contains some very interesting observations about extracting statistical features from images. The scalar results from the feature equations just derived are very dependant on how they are implemented and the kinds of textures they are implemented on. For example, textural and tonal information can both be present in an image or either one can dominate the other. If texture is the dominant information in the small area, then this area has a wide variety of discrete tonal features. However, if the number of distinguishable discrete tonal features declines, then the tonal properties dominate. Three variables exhibit control over the values just described. They are:

- The size of the image window under investigation.
- The relative sizes of the discrete tonal features.
- The spatial distribution of distinguishable discrete tonal features.

The importance of developing classification and detection algorithms that include both spectral and spatial information cannot be understated. In most surface mapping applications, only spectral features are used. Pixel classifications are based solely on their "color", as with the generic GML classification algorithm described earlier. The spatial relationship to neighboring pixels is irrelevant. In fact, even if all of the pixels in the image were randomly distributed, each individual pixel would still be classified to the exact same category as it would have been in the original image. Kiang (2002) does a

comparison of classifying an Ikonos image based solely on spectral data to classification which included both spectral and textural information from GLCMs. Compared to per-pixel classification with only spectral measures, the textural features can improve the accuracy by as much as 7% (or 12% relatively). Also, including the mean and standard deviation of the pixel values under the GLCM window can improve the accuracy by as much as 25% (or 43% relatively).

The combination of GML and GLCM algorithms will provide measures of both the general spectral and spatial accuracy of the synthetic imagery. The real goal of the Army MURI is to accomplish concealed target detection and therefore the final metric will be to analyze both the truth and DIRSIG imagery with the RX algorithm.

2.3.4 RX

The RX algorithm was originally described by Reed and Yu (1990) and also Reed et al. (1989). This algorithm relies on the assumption that the image clutter can be described as a Gaussian random process with a fast spatially varying mean and a more slowly varying covariance. The RX algorithm uses a combination of spatial and spectral information to detect targets in an image through a convolution-like process. The signature of the target is considered to be additive to the signature of the background clutter in the calculations. An important aspect of the algorithm is that it has a constant false alarm rate (CFAR). This property allows the user to have a much better understanding of the detection results as the RX kernel passes over the image. As the RX kernel, which will be described shortly, moves about the image, the statistics underneath it change. Without the CFAR normalization it would be very difficult to have any confidence in the stability of the detection rates throughout the image.

To begin the explanation of RX, let the spectral values of each pixel be represented

by Equation 2.17, where DC is the digital count in that spectral band and l is the number of bands in the image cube.

$$\mathbf{x} = \begin{pmatrix} DC_1 \\ DC_2 \\ \vdots \\ DC_l \end{pmatrix} \tag{2.17}$$

Then, let the pixels under the RX kernel be represented by Equation 2.18. The sub-image defined by the kernel represents a square matrix of N pixels centered at the pixel of interest.

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \dots & \mathbf{x}_N \end{pmatrix} \tag{2.18}$$

The spatial shape of the target as it would be found under the kernel is expressed as Equation 2.19. The values in the spatial shape vector in that equation are normalized such that $\mathbf{S^T S} = 1$.

$$\mathbf{S^T} = \begin{pmatrix} s_1 & s_2 & \dots & s_N \end{pmatrix} \tag{2.19}$$

From here, the algorithm is designed to go in two directions. One implementation deals with a known target spectral signature, $\mathbf{b^T}$, shown in Equation 2.20.

$$\mathbf{b^T} = \begin{pmatrix} b_1 & b_2 & \dots & b_N \end{pmatrix} \tag{2.20}$$

The other implementation allows the user to detect anomalies, or signatures that are significantly statistically separated from the local background clutter. To make the RX algorithm an anomaly detector substitute Equation 2.21 for $\mathbf{b}$.

$$\hat{\mathbf{b}} = \mathbf{XS} \tag{2.21}$$

Without derivation (which can be found in either of the two references mentioned at the beginning of this section) the formula for the CFAR RX algorithm with a known target spectra is shown in Equation 2.22. In that equation, $\mathbf{M} = \frac{1}{N}(\mathbf{X} - \bar{\mathbf{X}})(\mathbf{X} - \bar{\mathbf{X}})^T$, is the covariance of the background clutter around the pixel of interest. Substituting Equation 2.21 into Equation 2.22 yields the anomaly detection version of the algorithm. Reed et al. (1989) showed how the two versions of the algorithm could be used in tandem. There, the anomaly detector was run first and the spectral signature of the pixel with the strongest response was used as $\mathbf{b}$ in the matched filter version to locate all targets in the image with that spectral structure.

$$r(\mathbf{x}) = \frac{\left(\mathbf{b}^T\mathbf{M}^{-1}\mathbf{XS}\right)^2}{\left[1 - \frac{1}{N}(\mathbf{XS})^T\mathbf{M}^{-1}(\mathbf{XS})\right]\left[\mathbf{b}^T\mathbf{M}^{-1}\mathbf{b}\right]} \tag{2.22}$$

Furthermore, two hypotheses are being tested during each calculation of $r(\mathbf{x})$. The first hypothesis, H_0 is defined as the pixel of interest containing only background, or $\mathbf{x}(n) = \mathbf{x}^0(n)$. The second hypothesis is that the target signature is present at the correct orientation, or $\mathbf{x}(n) = \mathbf{x}^0(n) + \mathbf{b}s(n)$. To implement that equation, the user must define some threshold value, r_0. If the value of $r(\mathbf{x})$ exceeds r_0 then it is assumed that hypothesis H_1 is true, if $r(\mathbf{x}) < r_0$ then H_0 is true. $\mathbf{x}^0(n)$ is the vector of residual clutter noise-only processes.

The four metrics described in the previous sections, ROC, GML, GLCM, and RX all compliment each other well for the purpose of this research. ROC will provide a baseline measure of radiometric accuracy. GML classification covers the generic spectral quality of the synthetic imagery. The GLCM texture metrics will help determine if the model

is correctly recreating the spatial variability that occurs in the real world. Finally, since RX is a target/anomaly detection algorithm, it suits the overall purpose of the MURI project well. The next section will describe how some other validation efforts have been conducted and the relevant issues that were faced during their execution.

2.3.5 Other Validation Efforts

It is important to include a brief description of some other validation efforts that have been done that are similar to the one being undertaken in this research. This section contains an overview of the experiments and a summary of some of the problems that were encountered so that similar misfortunes may be avoided.

One of the closest in objective was conducted by a team of scientists from the United Kingdom. The set of experiments was called *Trial SNAPSHOT*, which was designed to test the fidelity of the CAMEO-SIM model. The overall details and design of the experiment can be found in Mitcell et al. (2000), and an analysis of the validation effort and its results can be found in Mitchell et al. (2000); Mitcell et al. (2002). The goal of Trial SNAPSHOT was to capture detailed information about a single, complex real-world site. The data was acquired across the spectrum from $0.4 - 14\mu m$ using visible band imaging spectroradiometers, telespecroradiometers (visible, NIR, SWIR, and LWIR), calibrated color cameras, broadband SWIR and LWIR imagers and contact reflectance measurement equipment. Seven instruments were used during the trial. They included:

- Surface Optics Real Time Imaging Spectrometer (RTSIR)^{TM} – visible and near infrared imagery
- Spectracube^{TM} imaging specroradiometer – visible and near infrared imagery

- Kodak DCS 420 digital camera – visible band imagery
- AGEMA^{TM} imaging infrared radiometer – 3-5 and 8-12 micron band imagery
- GER 3700^{TM} telespectroradiometer – 0.4-2.5 micron spectral reflectance measurements of different surfaces
- Spectrascan^{TM} telespectroradiometer – visible band spectral radiance measurements of white standard panel
- LMT luminance meter – luminance measurements of white standard panel to monitor changes in illumination

The site selected for acquiring the imagery consisted of primarily south facing mixed coniferous and deciduous woodland, but also contained uncultivated grassland and tracks, very much like the MicroScene area at RIT. Controls were placed in the scene for ground truth measurements. These included six painted panels - 3 matte panels and 3 gloss panels, a landrover, and shaped wooden cut-outs with different paint and camouflage material finishes to simulate men. Meteorological data was also collected throughout the trials. Beginning one day prior to measurements, wind speed, wind direction, average dry temperature, average wet bulb temperature, relative humidity, average solar radiance, average net radiance, barometric pressure and soil temperature were all recorded.

Three different real and synthetic pairs of the scene were created. The first pair was imaged with no targets, the second contained the landrover partially obscured, and the third pair showed the landrover completely out in the open (Mitchell et al., 2000). To validate the results of the CAMEO-SIM imagery three types of analysis were conducted. The first analysis tool selected was the use of higher-order statistics (specifically fourth order). The second was a model based on the human visual system.

Finally, multiresolution feature extraction was performed using difference of gaussian (DoG) filters. These filters were chosen because of their importance in nearly all image processing systems, including human vision and autonomous seekers.

The higher order statistics (HOS) metric was chosen because of its inclusion of the images' phase spectra, which provides clues about the shape and relative positions of features in the image. The traditional mathematics was slightly modified so that local statistics could be garnered from pairs of images being compared. Difference maps were created which showed large values around regions of the scene containing the landrover target implying that its modeling was in some way inadequate. The determination was that the target, with its sharp edges, contained too much fine-scale coherence relative to the real images. This drove an upgrade to the sensor model as it was not sufficiently noisy at the finer scales. In the end, the HOS produced similar results between real and synthetic versions of the same scene and different version of the real scene taken at different times. Although, introducing the landrover into the real scene did not produce the significant discrepancies that occurred when it was introduced into the synthetic scene.

As described in Nyberg and Bohman (2001), multidimensional analysis can provide target detection clues because man-made objects tend to be non-fractal, or finer details do not emerge as one examines higher spatial resolutions. The SNAPSHOT team examined this phenomenon in Filbee et al. (2002) and Mitchell et al. (2000); Mitcell et al. (2002) using four parameters that they considered to be important to match between real and synthetic textures:

1. A measure of overall clutter strength, pc.

2. A measure of image smoothness (spatial correlation called the self-similarity parameter), k.

3. A measure of overall clutter density, *a*.

4. A measure of clutter uniformity, *d*.

The parameters can be plotted as a scatter plot to provide a comparison between the real and synthetic scenes. Overlapping points suggest that the synthetic image closely mimics the real image. Interestingly, Mitchell et al. (2000) points out that while it is generally more important to know *why* a simulation went wrong, the scatter plots can be useful for giving an indication of by *how much* it went wrong. The results of the analysis were that the simulation was under-correlated spatially. This would lead to an overestimate in the difficulty of detecting small-scale targets or camouflage structure compared to large-scale information, which is very important to know when attempting to train ATR algorithms.

Validations of earlier versions of the DIRSIG model have also been accomplished (White, 1996; Kraska, 1996; Joseph, 1998; Mason et al., 1994). The work was generally done as a validation of the radiometry model of DIRSIG. White and Kraska worked in tandem to produce a recreation of a well studied 1km x 1km scene in northwestern Rochester. White validated the visible portion of the model, while Kraska's focus was on the thermal. The majority of their time was spent on the construction of the model. Digital elevation maps, city plans, and material samples were all gathered together to replicate a truth image of the area taken with Bendix Line Scanner in the $8 - 14\mu m$ band, an Inframetrics Model 600 IR Imaging Radiometer in the $8 - 14\mu m$ band, and a Kodak Digital Camera System (DCS) - 420 for the visible region. The images were taken both at nadir, from a low flying aircraft, and oblique viewing angles at resolutions ranging from approximately 1 to 3 feet.

The approach used by both individuals was to spatially register the real and synthetic images and produce difference metrics based on rank-order correlation and root-

mean squared (RMS) error. RMS error is a difference metric that requires the two images to be exactly spatially registered. This is inappropriate for this research for reasons mentioned in the beginning of Section 2.3. Their result showed that DIRSIG produced accurate radiometric data across the spectrum from $.4 - 14\mu m$. Although, much work was done to produce geometric variation, qualitatively, the synthetic imagery was very flat. The size of the scene being recreated was considered very large at the time, and time and technology constraints created an environment where little detail could be provided to the model in terms of the number of materials in the scene, texturing could not be added, and there was a lack of mixed pixels. Additionally, Kraska's thermal truth images were taken without a great deal of meteorological ground truth which resulted in a trial and error approach for modifying the MODTRAN inputs. Both Kraska (1996) and Joseph (1998) point out the importance of accurate meteorological input data to the overall performance of MODTRAN.

Joseph (1998) continued Kraska and White's work by conducting a broad validation of DIRSIG from $0.4\mu m$ to $14\mu m$ wavelengths. In that work, he simulated an Army data collection called the Western Rainbow study, which was designed to measure the spectral signatures of various vehicles and backgrounds under differing conditions of camouflage, concealment and lighting at three study sites using Daedalus MSS, Hyperspectral Digital Imagery Collection Experiment (HYDICE), and Spatially Enhanced Broadband Array Spectrograph System (SEBASS). Joseph consolidated the three study sites into one representative synthetic scene that was not meant to be spatially registered by any means, but to provide a means to perform ROC calculations and qualitative spectral comparisons.

Problems encountered by Joseph included undersampling, inadequate material database population for characterizing some scene objects, and a lack of overall detail. Specifi-

cally, his results showed that the synthetic vegetation class was habitually more radiant than the real data. The author attributed the problem with the rank-order assigned to the vegetation class to undersampling of the scene. Shadow and background effects were not being simulated adequately and so a mixed vegetation material type was substituted into the simulation to increase the variability of the desert foliage and better results occurred. The material database was also underpopulated with regard to some of the object material types.

Joseph also demonstrated the importance of properly attributing object facets with the correct material type. The emissivity data for a SCUD missile launcher was unavailable so the material data from an R20 vehicle was assigned to the synthetic SCUD with erroneous consequences. The proximity of MicroScene proved invaluable in this regard as any scene materials were easily accessible for measurement.

A lack of detail was the other issue Joseph faced. This problem has not only plagued him but also most every other DIRSIG validation effort. A high-resolution data collection effort has not been possible until now for a variety of reasons:

1. Computer processing has not been powerful enough.

2. Databases used have not contained the kind of variability characteristic of real-world materials

3. Proper texturing algorithms had not been implemented or validated

For these reasons it was prudent to conduct this DIRSIG validation effort for verifying exactly how capable this SIG tool is for modeling high-resolution scenes that were previously too complex for simulation.

2.4 Camouflage

Finally, it is necessary to discuss the various types and ways of concealing military vehicles from view. People have been using camouflage in some form or another from the beginning of human civilization. In fact, camouflage got its start long before humans ever thought of using it. The animal kingdom has been coming up with various ingenious ways for hiding prey from predators since as far back as science can probe. Also, many of the techniques that we use for concealment have been derived from the methods that plants and animals use. This section provides a brief overview of camouflage in general. Following that is a discussion of the tactics that the Army uses to conceal vehicles. That discussion is focused on the two kinds of concealment that are modeled in the MicroScene area: natural camouflage and camouflage netting.

2.4.1 Camouflage Overview

Until very recently, as far as history goes, camouflage was designed around two variables, color and pattern. The color was designed to mimic the colors of the naturally occurring backgrounds that a soldier may be hiding in. But, just because the colors are right doesn't render a soldier invisible. Too much of any one color, even if it does match the background, can serve to distinguish its wearer. Therefore, the pattern of the colors is also important. It must be designed to arrange the correct colors in such a way that their spatial frequency matches the spatial frequency of the background. This serves to disrupt the human visual system, which naturally categorizes things in the world as being separate objects because they exhibit continuous outlines. Camouflage disrupts the perceived continuity and hides its wearer from view even though they may actually be otherwise unobscured from sight.

As technology advanced, it was soon discovered that camouflage may help hide its

wearer from the human visual system, but may not provide any protection from other kinds of sensors. New sensors were designed to take in wavelengths of light that are undetectable to the human eye. These sensors, specifically thermal and SWIR types, were immediately capable of differentiating camouflaged targets from their background because of sharp thermal contrast or mismatched reflectance properties in the non-visible portion of the spectrum. Current upgrades to camouflage include, materials that exhibit similar spectral qualities to that of their respective backgrounds over a much broader range of wavelengths, engine systems that mask their thermal output, radar reflecting "stealth" materials, and even simple decoys.

2.4.2 Vehicle Camouflage

The two primary means of camouflage available for this research were natural vegetation and camouflage netting. Therefore, it is important to understand how the military utilizes these two concealment techniques in more detail so that they could be accurately modeled. According to the Department of the Army (1999), the field manual on Camouflage, Concealment and Decoys (CCD), forests generally provide the best type of natural screen against optical reconnaissance, especially if the crowns of the trees are wide enough to prevent aerial observation of the ground. The importance of shade in concealing vehicles was also mentioned. Whenever possible, branches and vines are used to temporarily conceal vehicles, equipment, and personnel. This is done by attaching vegetation to equipment with camouflage foliage brackets, spring clips, or more expedient means (such as plastic tie-wraps). When implementing this concealment technique the orientation of the leaf is also important as the underside of green vegetation is generally much lighter than the top.

The use of natural vegetation is mentioned as a temporary concealment technique

because cut foliage wilts and changes color rapidly. This is due to the loss of water and chlorophyll, the substance which gives vegetation is nice blending properties in the NIR. For this reason, coniferous vegetation is preferred to deciduous because it holds its chlorophyll for a longer period of time after being cut. All of this is not to exclude the use of dead vegetation, though. Dried grass, hay, straw and branches can all be used for texturing. It provides good blending qualities if part of the surrounding background vegetation is also dead.

The other type of camouflage to be used in the MicroScene collection is camouflage netting. The netting is generally available in three color patterns: desert, woodland, and arctic. Also, each side of the netting has a slightly different pattern to allow for seasonal variations. Most netting systems are modular and consist of a hexagon screen, a diamond-shaped screen, a support system, and a repair kit. Any number of screens can be joined together to cover objects or areas of varying sizes.

A basic camouflage net conceals in three different ways:

- Casts patterned shadows that breakup the characteristic outlines of an object.
- Dissipates infrared radiation.
- Simulates color and shadow patterns that are commonly found in a particular region.

In the field, netting is rarely used without supplemental camouflage. Cut foliage can be used to augment the concealment effect, although the same precautions regarding orientation and wilting mentioned earlier should be adhered to here also.

2.4.3 Camouflage in DIRSIG

Although it wasn't necessary to produce the virtual camouflage net that was made for the MicroScene model, Loyd and Sanders (2001) used a software package called SIM-NET. It may be a useful automated alternative to the manual technique described in Section 3.2.3.1. The program is designed to virtually drape a 2D fabric structure over a 3-D object. It begins with a triangular facet representation of the fabric. Mathematically, the facet vertices are simulated as point masses and the edges of the facets are assumed to be springs. The process begins with the 2D fabric in a completely flat state suspended above the object. Under the hood, the simulation is driving the fabric to its lowest potential energy configuration, which is why the assumptions about point masses and springs are made. All of the support structure of camouflage netting described in the Army field manual on CCD can also be included for very realistic looking camouflage.

Loyd explained that post-processing was necessary to replicate the texture of the incised vinyl used to garnish the camouflage netting. In an example from that paper, netting was draped over a SCUD missile Transporter Erector Launcher (TEL) that contained 2226 facets before being converted into 3-D geometry with much finer facets and then a subsequently higher facet count of about 105000. In the final version each facet was constrained to be no bigger than 0.1 m. A CAD program was used to randomly remove half of the facets to create netting geometry that contained holes. After adding the holes, the remaining facets were randomly shifted on all three axes by a margin of -1.5 to 1.5 cm with uniform probability to eliminate the artificial smoothness created by the simulation. The constraint parameters used in the facet size, hole creation, and vertex randomization were given no justification other than to show an example result that was qualitatively realistic. After all was finished, the result was

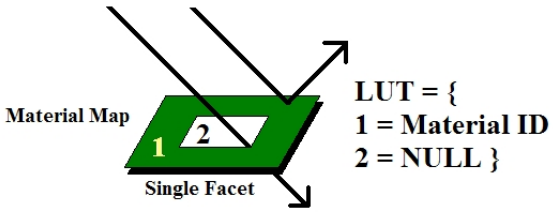


Figure 2.16: Diagram of DIRSIG's treatment of NULL material mappings.

saved in *.OBJ format for inclusion into DIRSIG.

One of the problems with the method just described is that the holes in the net needed to be cut from the model at the facet level. This is very difficult and it requires many additional unnecessary facets to achieve accurate results. DIRSIG users can produce the same results with fewer facets and less work by capitalizing on its hierarchical mapping capability. A material map can be assigned to the camouflage net model that contains an index color that is referenced as "NULL" in the map's look-up table. When DIRSIG casts a ray that hits the portion of a material mapped facet with a NULL material ID, the ray is allowed to pass through the facet as if it wasn't there. A graphical representation of the how DIRSIG treats NULL material mappings is shown in Figure 2.16. The capability reduces the modeling needs that would otherwise be required if each hole in the net was cut from the model on a facet basis.

This concludes the background and literature review portion of this research. This chapter provided an overview of many popular SIG tools available, including DIRSIG, RIT's own modeling software. It also described the four methods, ROC, GML, GLCM, and RX that are used to validate the synthetic imagery created by DIRSIG, in detail. Finally, a summary was given of camouflage and concealment techniques along with how to model those techniques. The next chapter discusses the process for conducting the truth image collection experiment, and the creation of the MicroScene model.

Chapter 3

Approach

A *continuous improvement* approach has been taken in the development of the synthetic MicroScene model for this research. This means that the model was validated in a cyclical fashion whereby the results were verified at certain milestones to assess the areas that needed improvement. The discussion in this section will essentially be a "How-to" guide for modeling some of the phenomenology that was created for this model. Where appropriate, some results will also be included to explain the author's intent for using a particular method. The goal is to give future readers solutions to many of the problems that required a good deal of time and research to accomplish. The first section of this chapter is devoted to the ground truth collection with the WASP and MISI sensors. The setup, data collected, problems faced, and the resulting imagery will all be discussed. The second section will detail the work completed to get the synthetic model of MicroScene into a configuration suitable for evaluation as well as the development of a suitable synthetic MISI instrument.

3.1 The Experiment

3.1.1 Collection Overview

Figure 3.1 provides a complete overview of the locations of the targets in MicroScene as well as the locations of the sensors. The placement of those objects was carefully selected so as to reduce the amount of buildings and open space in the background, and to optimize the placement of the primary targets in the scene.

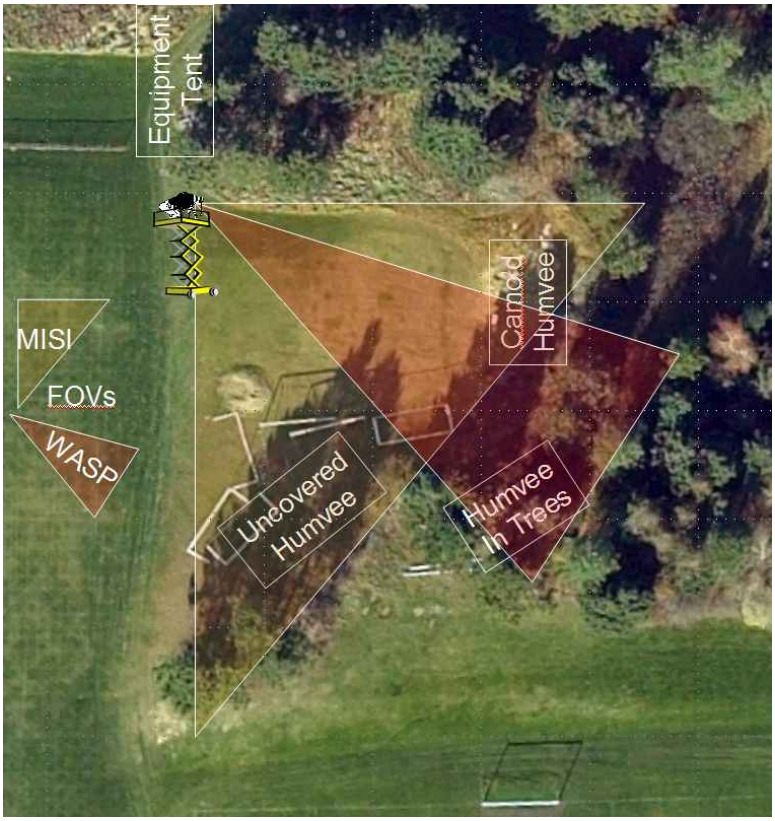


Figure 3.1: Sensor and target locations in MicroScene.

The scene was imaged with RIT's MISI and WASP instruments. They were placed on top of a scissor cart (see Figure 3.2) in the scene at the location depicted in Figure

3.1. The sensors were lifted with the scissor cart to 50 feet above the ground. The base of the cart was approximately 100 feet from the center of the target locations. The resolution of the sensors at this distance ranged from approximately .25 inches (WASP color camera) to 3 inches (LWIR detector on MISI).



Figure 3.2: WASP and MISI positioned on a scissor cart.

Three military Humvees were supplied through the generous assistance of the US Marines of the 8th Tank Battalion in Rochester, NY. Each Humvee already had a level

of standard woodland camouflage paint applied to it, so this form of camouflage was included in all configurations. One of the Humvees is shown in 3.3. All three are essentially the same.



Figure 3.3: Humvee target

To simulate the different ways a vehicle may be hidden by trees, the vehicles were placed at three locations in the scene. The first placement was at the box labelled as "Uncovered Humvee" in Figure 3.1. This was done so that no trees or forest are in front the vehicle and only limited tree cover behind. The second level of concealment was accomplished by placing the vehicle in the location of the the box labelled "Humvee In Trees" so that the vehicle was surrounded by trees and only partially visible to the sensors. Finally, the last Humvee was placed at the location of the box labelled "Camo Humvee", where it was draped with woodland camouflage netting. The setup can be seen in Figure 3.4.

Cadre from the Army ROTC office at RIT were generous enough to assist in the

setup of the camouflage netting. Their field experience in the setup was invaluable and eliminated a large amount of time from the setup process. With their help, the camouflage was setup to the specifications set forth by the Army for camouflage deployment. Spreaders and stakes were used to create a domed effect around the camouflage to eliminate any distinguishing edges. Supplemental camouflage (i.e. twigs, grass, leaves, etc.) entwined in the net was not used for the sake of simplicity. The Humvee was placed in this configuration in the open so that a clear view could be obtained of the camouflage netting, its contours and also the shadowing created by the camouflage pattern.

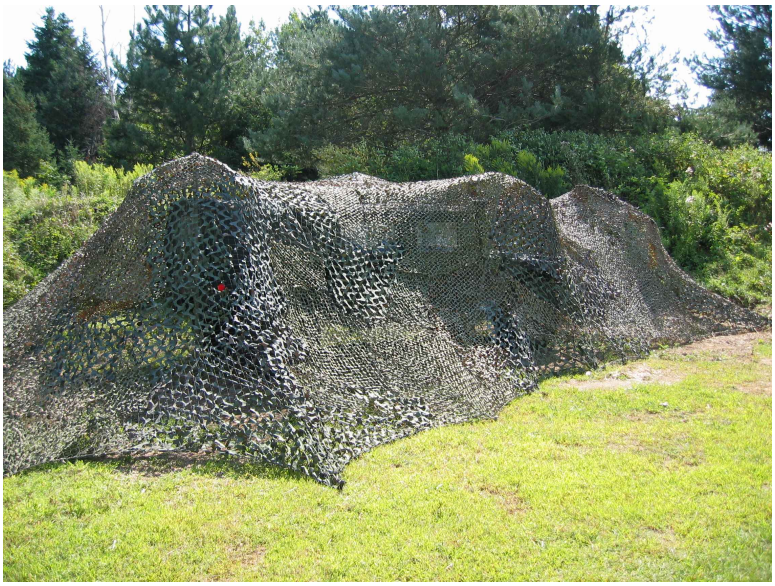


Figure 3.4: Humvee target under camouflage.

In addition to being imaged by MISI and WASP, the LCSS camouflage was imaged from underneath the camouflage with a standard digital camera. This image was taken when the sun was directly overhead to maximize the speckled shadow pattern cast on the Humvee from the netting. This image is shown in Figure 3.5.

Many other pieces of information were needed to correctly characterize the environ-



Figure 3.5: Complex shadows cast by the camouflage netting.

ment and the objects in the scene for the simulation to be completed. In addition to the image, many reflectance curves of the surrounding background were also collected to populate the material database. Also, detailed meteorological data was collected to categorize the effects of the weather. A number of tools were made available for doing weather measurements, such as surface temperature, wind speeds and direction, as well as humidity, at the time of the collect. A broadband measurement of the downwelled radiance was also collected. The following materials and sensors were used in the collection to gather the aforementioned data:

- ASD Spectroradiometer - spectral measurements from .4 to 2.5 μm
- SOC 400T FTIR - spectral measurements from 2 to 25 μm
- Portable Meteorological Station
  - wind speed

  - temperature
  - humidity
  - data logging capability

- 1 white control tarp
- 1 black control tarp
- 3 tile control panels: black, gray, white

This collection was not done solely for the purpose of the completion of this project. Other research, primarily dealing with the detection of buried land mines as well as various projects within the CIS department, were also being conducted. While their data may not be important to this project, some of instruments used to obtain that data provided additional small target clutter that was included in the synthetic model of the area. The following is a list of other items that appear in the scene that support those other objectives.

- 4 simulant land mines buried to a depth of approximately 5cm
- 4 surrogates for simulating surface land mines
- 1 3kw generator
- 2 blackbody control targets
  - CI Systems IR Radiation Source Controller
  - Techne TU-16A
- Staring IR thermocouples

- Contact thermocouples

Now that a general overview of the collection has been laid out, it is important to continue with a more detailed account of the the operations that were conducted and the timeline of events beginning with the pre-collection phase.

3.1.2 Pre-Collection Phase

Many issues needed to be resolved in the weeks prior to the experiment for it to occur smoothly. These issues ranged from logistical matters to equipment setup. Due to the unique nature of this data collection, a more in-depth overview of what went into its accomplishment is being laid out here to assist in any similar future endeavors.

The data of the experiment was chosen to begin on Monday, 25 August 2003. All of the equipment and personnel were scheduled to be on-hand for a full week to allow for adjustments due to weather conditions or other unforseen events. Two weeks or more prior to the event, weekly planning meetings were conducted, the scissor cart and Humvees were reserved, and a data logger, cables, wires, and thermocouples were purchased. Also, due to the unique implementation of the MISI sensor, a motorized turntable was built to control its movement through the scene.

The WASP instrument consists of four framing array cameras. Therefore, no additional setup was required beyond pointing them in the appropriate direction and taking the images. MISI is a line scanner and, therefore, required the extra attention. To simulate aircraft movement, a mechanical turntable was placed under MISI. A laptop computer controlled the turntable which rotated MISI through the scene to advance each new line at a constant rate of about 20Hz. Normally, line scanners scan in the horizontal direction (i.e. perpendicular to the flight line). Since the rotation of the turntable provided the horizontal translation of the scanner, the instrument needed

to be turned on its side so that the scan lines ran vertically through the scene.

At one week prior to the collection the following items were addressed. A high bay was reserved from the Center for Industrial Management Systems (CIMS), a department at RIT, for sensor storage and setup of the sensors on the scissor cart. RIT's Facilities Management team assisted by providing a 21Kw generator and gasoline, all-weather tarps, a canopy tent, tables and chairs. Coordination with the soccer team as well as Campus Safety was also accomplished. A work schedule was created to ensure that collection was manned throughout the week and the area was cleared of debris that would unnecessarily clutter the environment. Finally, the computers were tested to ensure that they would be able to run the sensor remotely from the ground so that no one would have to be with them up in the lift.

The control tarps, resolution chart, 2 blackbody generators, thermocouples, a portable weather station, downwelled radiometers, the Humvees, camouflage, and MISI and WASP were all put in place on the morning of the collection. Other miscellaneous items that were obtained were liquid nitrogen for MISI, day and night clothing, mosquito repellant, food, and entertainment for the workers.

3.1.3 Data Collection

Data was collected every hour, on the hour, with both WASP and MISI positioned approximately 120 ft from the center of scene. Also, weather data and temperatures were collected every 5 minutes. The first official images of the scene were taken at 1900 on Monday the 25^{th} under sunny conditions with minimal cloud cover. The last images were taken at 0600 on Tuesday due to rain. Fortunately, the emergency tear down plan was implemented immediately and the sensors were wrapped in tarps and driven to a loading bay where they could be kept under cover until the weather cleared.

The team regrouped to start collecting imagery again Wednesday the 28^{th} by 1000. 30 collections were obtained through Thursday at 1500. The cloud cover was nearly non-existent on Thursday so it was decided to continue the collection and overlap some of the previous collections in the hopes of getting even better data. There were 43 "good" collections over the four day period for a total of 215 images between the two sensors. The data can be found on the CIS network in the /cis/static/dirs/MicroScene directory.

Unfortunately, most of the spectrometer data collected by MISI turned out to have a substantial amount of solar glare in the imagery. This problem was not observed during the collection and wasn't identified until post-analysis. Since the collection, a shroud was created to be placed around the instrument so that only light coming from the front of the sensor will reach the detector in any similar collection configurations.

For this reason, the validation was split between two images to encompass the VIS/NIR spectrometer data. An image taken at 1000 did not contain the glare in 28 bands of the visible portion from .4 to .7 um, but did have glare in the NIR portion of the spectrum. The opposite was true for an image taken at 1800. In this image, 15 bands of usable data from .73 to .99 um was used and so these two data sets became the main drivers in the validation. Figures 3.6, 3.7 and 3.8 are sample images from the Terrapix visible camera, a visible MISI image from 1000 and a NIR band from the MISI image taken at 1800, respectively. The design of the synthetic version of MicroScene that has been built to represent this truth imagery will be the focus of the rest of this chapter.



Figure 3.6: Terrapix image taken at 1000.



Figure 3.7: Color image from MISI's vis spectrometer taken at 1000.

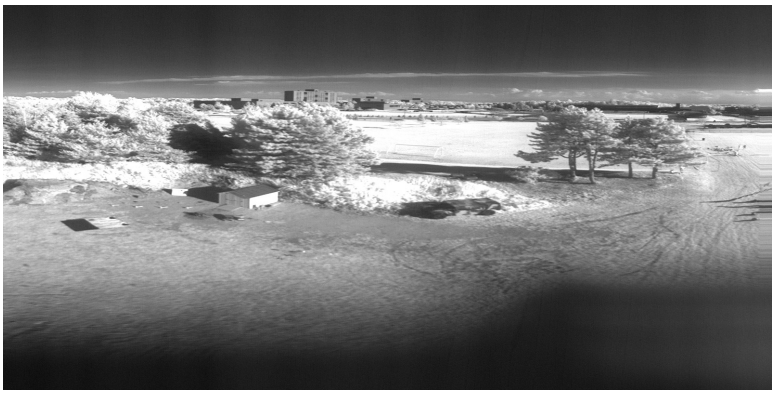


Figure 3.8: Single band from MISI's NIR spectrometer taken at 1800.

3.2 Virtual MicroScene Construction

Virtual scene construction requires the merging of a great many individual pieces that all must come together to produce as realistic a scene as possible. To address the spatial aspects of the scene, the terrain must be synthesized, 3-D models of all of objects must be created, various maps must also be built to provide realistic variability, and then each of those pieces needs to be appropriately placed in the virtual world relative to one another. The spatial qualities of imagery are also impacted by the characteristics of the sensor used to produce that imagery. The focal length, detector size, camera type (e.g. framing array or line scanner) and optical properties impart distinct effects into the imagery.

The spectral aspects are perhaps more important than the spatial aspects for this type of scene. Accurate spectral signatures require detailed collection of material reflectance spectra, accurate atmospheric estimation, and even the inclusion of bi-directional effects for non-Lambertian material types. Again, the sensor plays a major role in the resulting spectral signatures in the scene. Detailed knowledge of the spectral response function of the sensor must be known so that it can be used to convolve the DIRSIG derived entrance aperture radiance in the same manner as the corresponding physical sensor.

This section will describe the process that was used to develop the MicroScene model into its current configuration. It begins with an overview of how the terrain was created. Next, the modeling and placement of the scene vegetation will be described. Then, the process for making the manmade object models will be discussed. Finally, an overview will be presented of the creation of the virtual MISI sensor.

3.2.1 Synthetic Terrain Creation

The first step in synthetic scene creation is producing the terrain for the objects in the scene to rest on. The MicroScene terrain map is derived from a survey of the actual MicroScene area. A grayscale .pgm image was created from that survey which describes the layout of the land in an image format. The terrain model was created using four different graylevel values which corresponded to terrain heights in units of meters.

Similar to most terrain models developed at RIT, this image was passed through a locally developed IDL procedure that interpolated values between the four different elevations so that smooth transitions between elevations occurred. The problem with this approach for creating a high-resolution scene like this one was that there was very little control over the amount of smoothing and the smoothing was applied evenly throughout the image.

Rather than use the interpolating program, the DEM image was manually smoothed in an image editing package so that the terrain matched as closely as possible to the actual contours of the MicroScene area. This method of generating realistic surface elevation transitions was particularly necessary in the area of the scene just to the left of the shed (see Figure 3.6). When the actual shed was built, a large amount of dirt was excavated from the hillside to even out the ground. Special attention was placed on the area where the dirt was taken from because it had a sharp elevation change and also on the general size and shape of the hill behind the objects in the scene. Once the image was finalized it was facetized using a UNIX based program called **pgm2gdb**. The specific command line used in its creation is shown here for reference in case the model is ever updated. These parameters are essential for ensuring that a new model would be at the same height and have the same ground sample distance (GSD). (Ientilucci, 2003) provides a more detailed step-by-step overview of creating terrain models in this

way, as well as the syntax for using the **pgm2gdb** utility.

```
pgm2gdb NewMicrosceneTerrain 10000 1 0 0 158 .288 .0157
```

A flat extension to the terrain model was created in addition to the detailed terrain model for the portion of the scene immediately in front of the imagers. This plate extends about a 1000 meters into the distance and the background objects, such as the buildings and more trees, rest on it. The oblique viewing angle of the sensors required that this addition be made so that the virtual landscape appears to extend all the way to the horizon.

Once the facetized terrain model was generated it was overlaid with three different mapping images that are designed to induce realistic variability. This is important because it is done without the need for increasing the number of facets used to define the geometry of the terrain or varying the material properties of the terrain on a facet-by-facet basis. The final terrain model combines material, texture and bump maps.

A material map is used to distinguish different material types within an object. In this scene, it is primarily used to create the transition between the grassy region to the dirt area in front of the shed. The first approach taken to accomplish this was to use the spray brush feature in a image editing software package to randomly distribute the color that identified dirt in the map across the transition. It was very obvious that this was not producing the kind of transition seen in the truth imagery. The transition was not simply random like the spray brush was creating, but had some definite spatial structure as well.

The next approach was to take a picture of the transition region with a digital camera and import that image into the editing software. Since the dirt was lighter than the grass in the images, it was supposed that the two could be separated using thresholding techniques. This turned out to be the case. After adjusting the thresholding level to the

desired amount, the transition region was cropped out of the image and then pasted into the material map image. The black and white colors of the thresholded image were reassigned to the material IDs for grass and dirt. Then, they were interactively arranged along the transition boundary using cutting, pasting, and rotation to recreate the same spatially structured transition seen in the actual MicroScene area. Figure 3.9 shows the resulting transition region cropped from a DIRSIG rendering of the area.



Figure 3.9: Transition region in synthetic MicroScene.

A texture map is used to enhance the variability within an individual material type. It imparts variability into the grass and dirt in this model. Initially, a one-foot GSD texture image was used for the grass, but it was obviously not at a high enough level of detail to be used for these high-resolution images. The one-foot imagery was very blocky and looked artificial. Therefore, an approximately three-inch GSD Landcare image of the scene was obtained. This image was registered to the DEM image that produces the 3-D terrain model by using ENVI's image registration tools.

Finally, a bump map is an image that is used by DIRSIG to characterize the amount of deflection that should be added to an incident ray that impacts the flat facet surfaces. The bump mapping effect was used also on both the grass and dirt regions. It was created in Photoshop using its cloud generating texture package. The "perlin" texture

was selected and an entire image was made of that texture. Various scaling parameters were assigned to the map until it produced an effect that seemed realistic.

There is also one minor quirk about the model that should be pointed out here. This model was built with the positive y direction pointing north, in terms of DIRSIG world coordinates. It was made this way because the overhead imagery that is applied to the terrain was produced in this manner. Unfortunately, it wasn't realized until much later in the development process that DIRSIG uses the positive x direction as its basis for true north. The result is that the sun location is not accurate.

A few things needed to be done to correct the location of the sun beginning with the use of a special tag called "USE_ALT_EPHEMERIS_GEOMETRY". For this model that tag must be placed in the options section of the configuration file and set to "TRUE". This tag is supposed to force DIRSIG to use the positive y direction as north. Unfortunately, this did not work exactly as planned and so a more brute force method was used to override the solar location completely. Prior to running DIRSIG the user can set a UNIX environment variable called SUN_ANGLES with the solar declination and azimuth. For example, the UNIX command used to set the sun correctly for the 1000 image is as follows:

```
setenv SUN_ANGLES ‘‘53.318248 68.070033’’
```

To produce the correct sun location, those parameters must be used in conjunction with the "USE_ALT_EPHEMERIS_GEOMETRY" tag.

There is one drawback to this method, though. It only works in the VIS-NIR region of the spectrum. Updating this environment variable does not affect how the sun is positioned for the thermal regions with respect to the dynamic model. The user is restricted to either letting DIRSIG position the sun based on the positive x axis as north or setting the USE_ALT_EPHEMERIS_GEOMETRY option to TRUE so that

the positive y axis is north. A utility was created that allowed a user to rotate all objects in a DIRSIG .ODB file. While the utility worked for the objects in the scene, unfortunately all of the texture and material maps were unable to be rotated. For this research, the SUN_ANGLES override was sufficient as the VIS-NIR region is all that is being validated. In the future this problem is projected to be completely remedied through features unique to the next generation of DIRSIG. DIRSIG 4 will allow the user more control of the sun location than is currently available in DIRSIG 3. The capability in the new version will eliminate the thermal issue.

3.2.2 Terrain Vegetation

This section contains a description of the various methods that were either attempted or used in the current version of the scene for creating the background vegetation. The vegetation was by far the most difficult aspect of the scene to make look realistic both spatially and spectrally. The high-resolution requirements of the scene are in direct conflict with the memory and computational constraints imposed by the systems being used to model these scenes. The trees and the 3-D grassy region on the hill need to contain hundreds of thousands of facets to get the necessary level of detail. Making changes and updates proved to be a slow and difficult process because of the computer power limitations. The reader should understand this while going through this section and realize the necessary trade-offs of each of the methods described before undertaking a similar project.

3.2.2.1 3-D Grass

In the proposal for this project an idea was bantered about that relied on a series of shells, similar to the models used for the camouflage netting that would be molded to

the hillside and hovered just above it at various increments. Then a texture map with imagery of the plantlife and a material map with the inclusion of a NULL material that allowed rays to pass through to under layers would be placed on the shells. The idea was that this configuration would be able to relatively closely simulate the variability and appearance of the actual plantlife with many fewer facets than would be required to model individual plants.

It turned out that modeling the actual plantlife was not as difficult, and it could be done without taking up a large amount of memory. To begin the terrain model needed to be imported into Rhino CAD. Then, the **Drape** command was used to create a NURBS surface[1] that matched the contours of the terrain model's triangular mesh.

At this point, a 3-D model of a flat patch of grass was imported into Rhino. The patch of grass was originally very square, made of long, thin facets, contained roughly 20000 of those facets, and only covered an area of about 7x4 feet. It was randomized and redrawn so that there were no hard linear edges at the boarders of the patch and the total area increased to 18x10 feet. 18 feet was about the distance from the base of the hill to where trees began growing. The grass patch is used repeatedly across the hill side, so the number of facets needed to be reduced to about 10000 total.

The reduction in the number of facets helped speed up the modeling process and also required much less memory than if the original patch had been used. Unfortunately, the new patch was much less dense than the original and so the grass facets were slanted backward from the sensor position so that more surface area of each facet could be seen by the sensors. A second patch of grass was created from the first and it was rotated by 180 degrees. The two patches of grass were alternatingly molded to the contours

[1]NURBS stands for non-uniform rational B-spline. It is a mathematical way of defining curves, surfaces, and solids. Rhino is built to handle objects described in this way much more efficient than facetized mesh objects.

of the hillside by using the **Flow** command to shape the grass patches along *isoparms* generated from the NURBS terrain model using the **ExtractIsoparm** function.

The grass model was saved in .OBJ format for importation into Bulldozer. Then, a model of a small clump of grass was used to patch up any holes or trouble spots left over from the modeling. Bulldozer's ability to randomly assign rotation values was used to assist in the randomization of the placement of these patches. A few instances of a model for goldenrod were also added to the grass.

The result of this process can be seen in the top image of Figure 3.10. A qualitative analysis of the figure shows that the grass is pretty good. While it does look somewhat more like a flowing wheat field than the more irregular vegetation along the hillside shown in the bottom image, the results were encouraging.

Unfortunately, this method generated grass that looked good only from the direction of the MISI sensor. From the back, the grass was practically transparent because all of the facets were facing nearly parallel to that line of sight. Since this model is destined to become the basis for wider research than simply this project, a different model needed to be made.

There is a freeware 3-D CAD package called Blender. Blender is the result of a compilation of a decade of work done by various individuals on the software's open source code. The package can be downloaded at http://www.blender3d.com. A blender module, called Fiber Version 2, was imported into blender to create a better version of the 3-D grass region on the MicroScene hill. Fiber Version 2 can be obtained at http://oregonstate.edu/∼dennisa/Blender/Fiber2/. It is generally used to create long strand-like objects like, hair, grass, or fibers.

To ensure the grass model followed the contours of the terrain model, the facetized terrain was imported into Blender. Then, all of the facets were deleted in the terrain

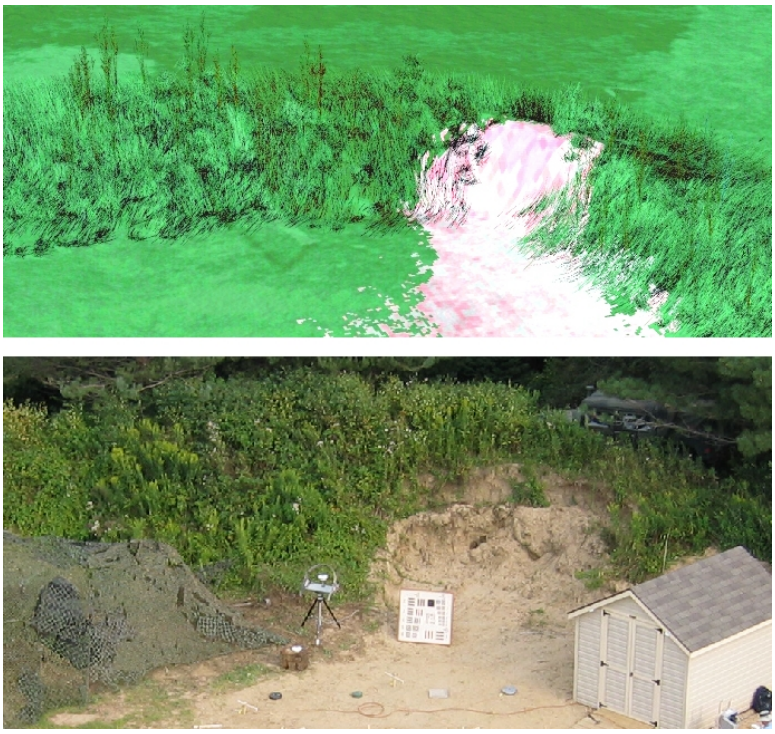


Figure 3.10: Top: Grass model rendered in DIRSIG – Bottom: Digital image of MicroScene

except for the ones on the hill where the grass was to be "grown". This process was facilitated by importing the material map used to create the dirt/grass transition. The material map was laid over the terrain and that allowed for the location of the excavated dirt region to be deleted from the grass growing software more accurately. The Fiber Version 2 software allows the user to set a series of options to obtain the desired grass effect. Height, blade thickness, bending, randomization and density were among the options that could be changed. Again, realism was balanced with the facet count to keep the model within memory requirements. The results are shown in Figure 3.11. The grass model shown in that figure is the one used int eh latest version of MicroScene.

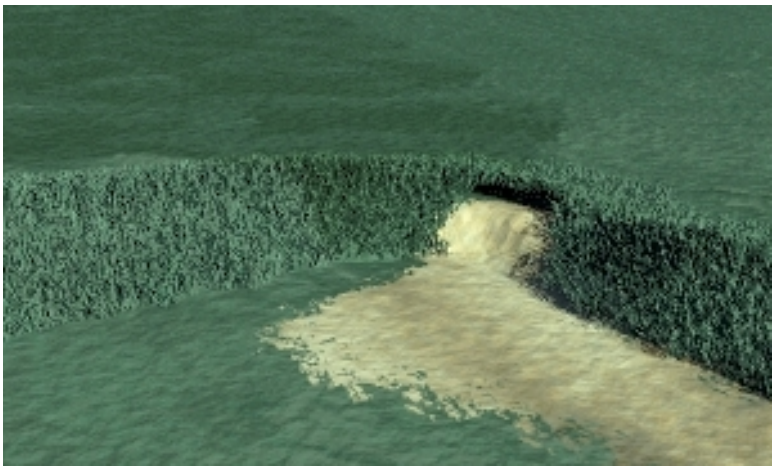


Figure 3.11: Blender generated grass model rendered in DIRSIG.

3.2.2.2 Semi-Automated Tree Placement

The current MicroScene model is populated with four different kinds of trees that were all produced using the TreePro vegetation modeling software. Two are deciduous and two are coniferous. The deciduous trees are primarily in the background and cannot be readily seen in the truth imagery. Coniferous trees dominate the portion of the forest that is visible to the MISI and WASP sensors. Therefore, a great deal of time was spent trying to find ways to make the forest look realistic in variability, while keeping the entire scene within the limitations of the computers available for rendering it.

Essentially, the modeling effort began by examining a host of tree models and then selecting the best ones for the scene. It was soon realized that it is difficult to separate one tree from the next at this resolution. So, only one pine tree model is used to generate all of the pine trees in the forest with the exception of the four trees that appear on the far right in the MISI truth imagery (see Figure 3.7 and 3.8). These four trees stood out so much that it was necessary to create a tree model specific to that type of tree.

One tree model was selected to represent each pine tree in the forest because DIRSIG allows for users to *instance* models into the scene at multiple locations without requiring more memory for each instance. The original model is loaded once and then it is referenced for each instance. Getting these trees in their appropriate locations was done with one of the most powerful and also one of the simplest tools created for this project.

That IDL program is called TreePlanter and its code can be found in Appendix B. The program works based on simple photogrammetry principles and it generates a .ODB text file in DIRSIG format. This utility was employed to plant the entire forest of trees behind the MicroScene area, and also the trees in the far background, without going through Bulldozer or doing the calculations by hand.

The main requirement of the program is an overhead image of the area that is registered to the DIRSIG coordinate system. To obtain an overhead of MicroScene to be used by the program, the WASP system was flown over the MicroScene area at an altitude of 4000 feet above the ground. Based on the Terrapix camera's $9\mu m$ detector size and $55mm$ focal length, the ground sample distance (GSD) of each pixel in the image is 0.2 meters. Photoshop was used, because of its image layering capability, to place black pixels at the centers of each tree in the forested areas in the image.

For simplicity sake, and to conserve computer memory, the virtual forest was designed using three different sizes of the single tree model. Three different image layers were created in the photo editing software. These three layers were made to represent small, medium and large versions of that tree model. For each tree in the forest, a non-white[2] pixel was placed in the appropriate layer, at that tree's center. After each tree

[2]Any non-white color can be used to make it easier for the user to differentiate which layer a tree has been planted in. The colors must be converted to 8-bit gray-scale with a pure white (255) background before being imported into the program.

was marked, the resulting three layers, one for each tree size, were saved as individual .TIF images (without the overhead image) on a solid white background.

Each image is kept separate so that they can be imported into the program individually and set the options appropriately for that tree size. TreePlanter.pro requires the user set a list of options befor each run. Table 3.1 lists all of those command line options and their meanings.

Variable	Meaning
XSCALE	GSD in x direction
YSCALE	GSD in y direction
XORIGIN	x pixel location of known DIRSIG coordinate
YORIGIN	y pixel location of known DIRSIG coordinate
XLOCATION	x location of origin pixel in DIRSIG coordinates
YLOCATION	y location of origin pixel in DIRSIG coordinates
ZLOCATION	z location of origin pixel in DIRSIG coordinates
OUTPUTFILENAME	name of output file
PATHNAME	.GDB file location on UNIX stations
SCALESTRING	x and y scale to adjust tree size properly Format example: SCALESTRING = ".6, .6"
HEIGHT	z scale factor to be output

Table 3.1: Summary of command line options for TreePlanter.pro.

As can be seen from the list of command line options in Table 3.1, the user must have some a priori information. For starters, the GSD of the overhead image must be known. The user must also know the x and y location of a reference point in the image that is also known in DIRSIG world coordinates. This point is crucial because it is used as the starting point for every distance calculation. The program converts the euclidian distance in pixels from that location in the image to every other pixel location and then converts the pixel distance to DIRSIG world coordinates based on the scale of each pixel. The program automatically rotates each object randomly about the z-axis and also randomly adds a value of $\pm$ .1 to the z scale value of each instance

for height variability.

The resulting .ODB file contains all the instances of the the object file being planted. It is completely configured to be copied into the main DIRSIG .ODB file. One of the more helpful additional features of the program is that each instance in the file has a comment directly above it that gives the location of the instance in the x and y coordinates of the overhead image. This is extremely useful if the user identifies prominent instances of objects and wants to make subsequent changes to enhance their accuracy. The location comments in the output files make this very easy. The trees that are located at the very edge of the forest, just behind the camouflaged Humvee and shed, were updated in this way.

Approximately 450 trees were planted in the virtual MicroScene using this process and it was completed in only 30-40 minutes. This process is great for planting large quantities of all kinds of generic objects, not just trees. For example, rocks or bushes could also be used.

The images in Figure 3.12 shows how accurate the technique was in recreating the forest region around MicroScene. The image on the left was taken with the panchromatic imager on WASP and the image on the right was rendered in DIRSIG. Spectral considerations are not important for comparing these two images. The WASP image was taken in the fall after the leaves had changed colors causing some trees to appear much brighter than they did in the summer. They are meant to show that if overhead imagery is available, then realistic forests can be populated with a fraction of the effort that has been put forward in the past.



(a) WASP overhead imagery (Fall). (b) DIRSIG overhead imagery (Summer).

Figure 3.12: Spatial comparison of WASP and DIRSIG overhead imagery.

3.2.2.3 Coniferous Modeling Issues

Once the spatial accuracy was acceptable, the next challenge was to correctly model the tree's spectral qualities. Again, resolution and memory requirements played a large role in the type of pine tree model that would be used. Also, considerations were made as to how light should be allowed to pass through the branches of these models.

TreePro allows a user to model coniferous trees in two different ways. The first method reduces the total number of facets required to model the tree by modeling at the branch level rather than the pine needle level. The branches are modeled as planar facets in the shape of a "+" sign that follows the contours of the branches. The resulting trees look realistic, but somewhat artificial at this resolution. A high-resolution example of this type of tree is shown in Figure 3.13.

Modeling pine trees in this way requires the user to make assumptions about how light should be allowed to pass through the branches. In reality light may pass right through a clump of pine needles unaffected, it may be reflected off of the needles in



Figure 3.13: TreePro coniferous tree modeled at the branch level.

different directions, or it may be affected by the transmissive properties of the needles. Unfortunately, with this type of model, incoming rays of light can only be affected in one way and the three processes just described must be consolidated into one material type for the DIRSIG database to calculate from.

The emissivity file used for the pine needle spectra was measured in the lab with a CARY 500 spectrometer. It was accomplished by taping individual pine needles together in a row such that as many gaps as possible were eliminated. The bunch was measured both horizontally and vertically in the instrument. The needles were also measured as a random cluster. The CARY had difficulty with measuring the random cluster because of its inherent non-uniformity. Also, the spectra from the pine needles that were taped together in both the horizontal and vertical orientations were nearly identical. So, that spectra was chosen to be used in the DIRSIG emissivity database.

The resulting spectral curves will be discussed in more detail in Section 4.2, but essentially, the resulting DIRSIG imagery that was modeled to match the 1000 truth imagery contained trees with much lower radiance counts. Also, a “halo” effect can

be seen in Figure 3.7 that makes the trees appear brighter as one looks further from their center. The halo effect could not be recreated with the pine trees modeled at the branch level. In real trees, the roundish needles reflect light in all directions, but the trees modeled at the branch level could only reflect light in one of four directions as dictated by the "+" sign design mentioned earlier.

Transmissive measurements of the pine needles were also made on the CARY 500, but it is very difficult to ensure that the transmission value produced by the measurement is not attributable to gaps between the pine needles in the clump. Therefore, an attempt was made to alleviate the lower radiance of the model by artificially increasing the transmission through the pine needle facets. Appropriate radiance levels could be attained if the transmission of the facets was scaled to unreasonable levels such that Kirchoff's Law was violated. In addition to it being physically inaccurate, this change that seemed to fix the 1000 imagery made the synthetic 1800 images completely inaccurate. The sun condition at 1800 was in back of the sensor and that created a situation where transmissive effects were not the primary driver of radiance values. The pine trees in the 1800 synthetic image produced radiance values more than five times greater than the truth imagery. These results necessitated the change to the trees modeled at the pine needle level.

A coniferous tree modeled in TreePro at the pine needle level is shown in Figure 3.14. This model contains considerably more facets than the branch level trees, but is still capable of being rendered on the machines available at RIT. These trees are the ones that are currently used in the model. The finer detail in these trees allows for a similar tree halo effect. Unfortunately, when rendered, the radiance values produced by them are still lower than the trees from the truth imagery. The problem was hypothesized to be an issue with the measured pine needle spectra. Though it wasn't specifically



Figure 3.14: TreePro coniferous tree modeled at the pine needle level.

noticed during the ground truth collection, newer buds on coniferous trees generally appear much brighter than the rest of the tree. After all else failed, while attempting to get the modeled trees to produce similar radiance spectra to the truth imagery, new pine needle buds were removed from a tree and were also measured on the CARY 500. The needles were measured in the same manner as described earlier and their general reflectance was slightly higher than the original measurements. A new emissivity file was created from the new measurement and the trees were re-rendered. Unfortunately, this also did not solve the problem as the radiance values were only about .2 radiance counts greater than when the original pine needle emissivity was used.

Other attempts were made to artificially reproduce the higher radiance values, such as making the needles very specular and simply increasing the reflectance of the needles manually. Certain pixels became brighter, but statistically the trees were more inaccurate because of these outliers. The average value was still much lower than the truth imagery suggested it should be.

The next possible step in trying to fix the tree models would be to increase the

tree's level of detail even more by replacing the flat pine needles that TreePro used with rounded ones. Perhaps this would allow for more light to be scattered in more directions and help to alleviate the heavily shadowed synthetic trees in the scene. The CIS department does not have a tree modeling program capable of producing this level of detail so trees of this kind could not be created for this project.

The phenomenon may also be associated with some microscopic mechanism peculiar to the pine needle itself. An interesting paper was discovered on modeling light scattering caused by human hair. In Marscher et al. (2003), detailed analysis of the microscopic surface of a human hair is determined as the cause of non-uniform scattering. When a model is used that takes this micro-roughness into account, then the modeled hair exhibits more realistic reflective properties. While the exact same mechanism described in that paper may not be the cause of the discrepancy seen in the pine trees of MicroScene, it does offer a glimpse of how microscopic interactions can manifest themselves at the macroscopic level.

3.2.3 3-D Models of Manmade Objects

Once the terrain and vegetation was finished, a large number of 3-D models needed to be created to populate the area with all of the same clutter as the truth imagery. 3-D facetized models of the various objects in the MicroScene area were built using the Rhinoceros CAD software package. Figure 3.15 shows a few of the more detailed CAD models that were built for this scene. From the top-left going clockwise they are: a downwelled radiometer, a shed, a toy wagon with an electric blackbody on it, a generator, a portable weather station, and finally a Humvee. That figure provides a sense of the level of detail that is trying to be attained in this model. These objects were placed in the scene using a combination of the Bulldozer scene placement tool and

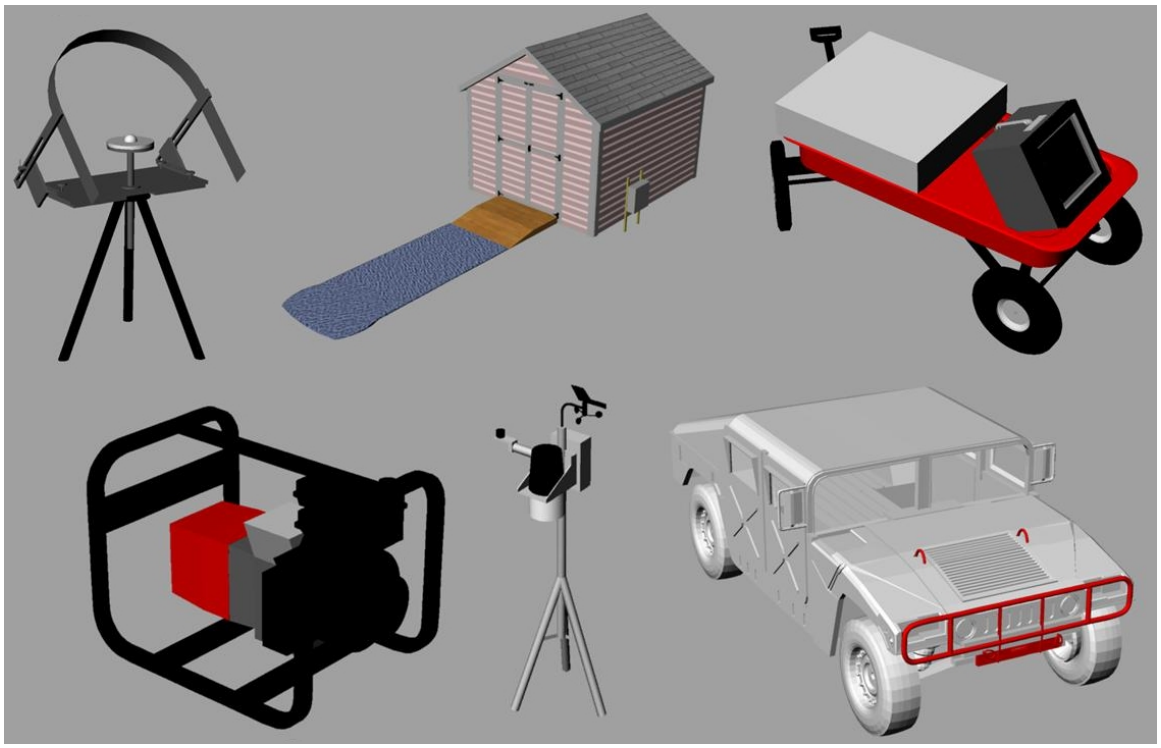


Figure 3.15: Various detailed models built for the virtual MicroScene.

empirical measurements taken during the actual collection.

3.2.3.1 The Humvees

The shed was the first model built because of its simplicity. The author had no experience with 3-D CAD software prior to this research and that model provided a relatively easy first production. Once the shed was complete, the next step was to tackle the various Humvee models and their different levels of concealment. Again, there were three different configurations that were imaged: out in the open, concealed by trees, and concealed by camouflage. Each of these configurations needed to be modeled and placed in the appropriate location in the scene.

Fortunately, the Center for Imaging Science had a 3-D model of a Humvee that was in a format suitable for being imported into Rhino. This model required a small amount of scaling to match it's dimensions to that of the Humvees used during the data collection. There are a few minor differences between this model and the actual Humvees. A crashbar on the front grill, hood hooks, and a slightly modified undercarriage were added to the base Humvee model.

That Humvee model could be directly used for two of the configurations mentioned earlier, but virtual support structure and draped netting were still required for the creation of the camouflaged configuration. The support structure was tackled first. A single support pole was created with the same dimensions as the ones used. The spreader at the top of the pole was even bowed slightly downward to represent the pressure that the actual net put on its support structure. The single pole was replicated four more times and all five pole models were placed in the correct locations and slant angles around the virtual vehicle. The Humvee model with the camouflage support structure is shown in Figure 3.16

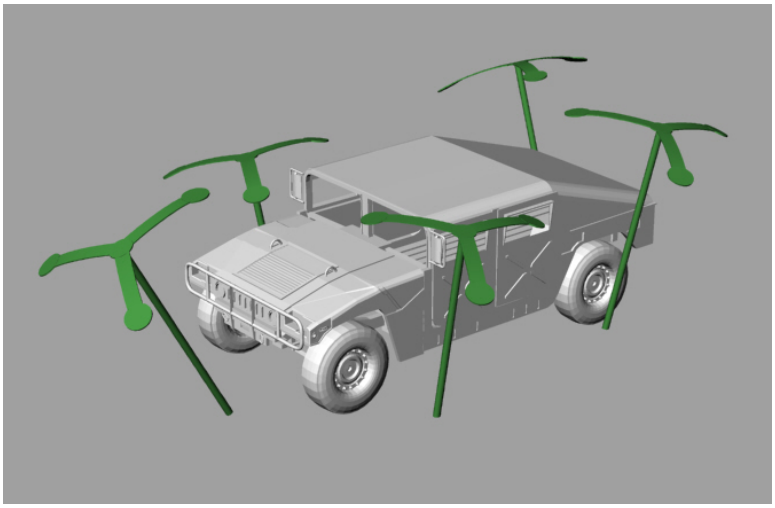


Figure 3.16: Humvee with support structure rendered in Rhino.

The proposal for this research spelled out the original plan for creating a realistically draped camouflage net by using the SIMNET software described in (Loyd and Sanders, 2001). That was the original plan because of the author's lack of knowledge at the time on using the modeling software. It turned out that creating realistically draped camouflage netting was not as difficult as expected. To begin its construction, eight sets of control points were placed from the ground on one side, over the support structure, to the ground on the other side of the vehicle. A curve was interpolated through each of these sets to form the contours of the netting. Careful consideration was taken to ensure that these lines did not fall below the plane of the spreaders. Once the curves were in place the **Loft** command was used to create a surface that encapsulated the vehicle just above the spreaders in the shape dictated by the control curves.

A significant problem was encountered when the model for the netting was rendered in DIRSIG. This problem is shown in Figure 3.17. This is a preview image of the camouflage netting rendered in DIRSIG. The obvious problem is that there are unintended holes in the netting along the seams of the facets.

All models that are drawn in Rhino are saved out at .OBJ files. The .OBJ files are also compatible with the Bulldozer utility. Bulldozer can import .OBJ files and convert them to .GDB files suitable for DIRSIG 3. After a great deal of troubleshooting, it was determined that the problem was occurring as a result of the way Rhino exported the facetized geometry. DIRSIG relies on triangular facets, but Rhino was exporting those facets as quadrilaterals... mostly non-coplanar quadrilaterals[3]. DIRSIG did not know how to handle the non-coplanar vertices of those facets and so interpolated them into planar facets. The result was the gaps that are evident along the facet seams in Figure

[3]Rhino 2.0 does allow the user to save files as raw triangles, but doing so eliminates all of the u, v texture coordinate information that will be needed when DIRSIG 4.0 is released for texture mapping as well as the material identification numbers.

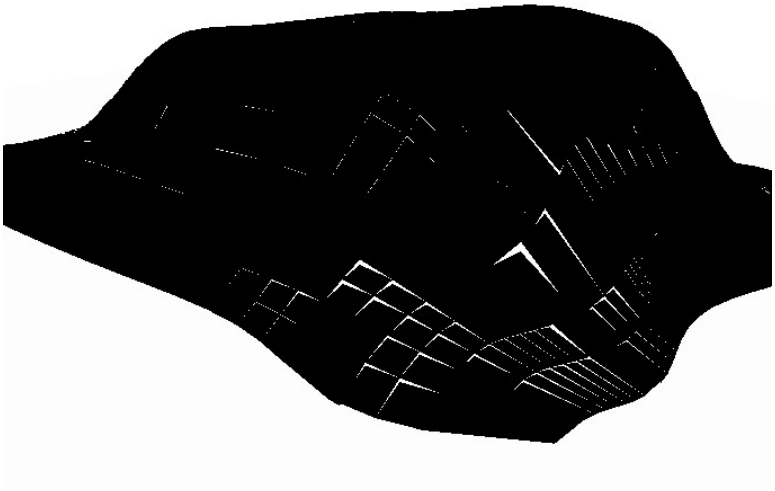


Figure 3.17: DIRSIG preview image of camouflage netting with non-coplanar quads.

3.17.

The solution to the problem was to triangulate all of the facets before rendering the model in DIRSIG. This process is sometimes referred to as *tessellation*. The basic idea is to split each quad into two triangles by bisecting it with a line between two opposite vertices. A local UNIX utility was developed called **munchquads** that converts non-coplanar quads in .GDB or .OBJ files into triangular facets without the loss of material properties or u, v coordinates. The resulting file can be saved as either a .GDB or .OBJ file.

All of the process just mentioned was accomplished only to get the general shape of the camouflage netting correct. Real netting contains different colors and it also is full of holes. DIRSIG's hierarchical mapping structure was instrumental for developing realistic virtual camouflage netting with all of these traits. A three color material map was created from a thresholded digital camera image of the net. The image

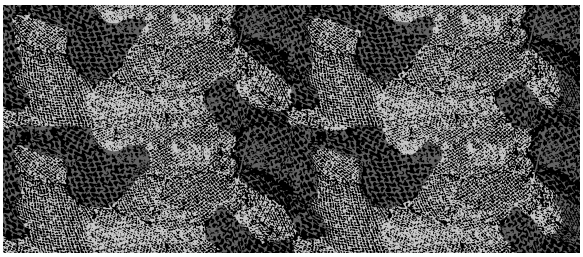


Figure 3.18: Threshold generated material map for camouflage netting.

was thresholded to a level that differentiated the net from the holes in the net in the same manner as described in Section 3.2.1 for creating the dirt/grass transition on the terrain. The white areas were recolored to represent the pattern of the net's two different camouflage colors. Rather than point to a measured material file, the black areas were assigned the NULL material ID. The camouflage material map is shown in Figure 3.18. The black areas in Figure 3.18 correspond to the areas assigned with the NULL ID tag in the material lookup table.

The dramatic results of this process are shown in Figure 3.19. The image on the left was taken with a standard digital camera from underneath the camouflage net around midday. The image on the right was created with DIRSIG by placing a similar "synthetic" camera under the synthetic net. The intricate shadow pattern on the vehicle is apparent in both images. It should be noted that DIRSIG's BRDF model, which helps to determine a more realistic background shape factor for each facet, was turned on for this image. The run time was dramatically increased, but the result was a much more realistic image. This side-by-side comparison is an example of the level of detail that can be achieved in DIRSIG.

There is more to this image than simply the shadow pattern, though. In general, when DIRSIG generates a scene, facets are considered to have only one side to speed up the rendering process. When creating a scene like this, it is important that the



(a) Digital camera image of the Humvee underneath camouflage.

(b) DIRSIG image of the Humvee underneath camouflage.

Figure 3.19: Truth vs. DIRSIG imagery comparison of a Humvee underneath camouflage netting.

facets that represent the net are visible from both sides because a ray that is allowed to pass through one side may intersect net on the other side. On the far right and left sides of both images in Figure 3.19, the underside of the camouflage net can be seen. The effect of the NULL material can also be seen in these regions as pockets of blue sky show through what would otherwise be a solid structure. The user must set the DOUBLE_SIDED tag in the .MAT file to TRUE to make the facets associated with a particular material ID double sided.

The effects of taking the extra time to render the image with the ENABLE_BRDF tag turned on in the DIRSIG configuration file are also evident in the figure. When images are rendered without the BRDF option turned on, shadows are generally much darker than they should be. Very little light shows up in regions hidden from direct sunlight because of the shape factor assumptions that DIRSIG makes to speed up rendering time. This is especially true in shadows casted by the camouflage net, the

crash bar, and in the wheel wells. Turning BRDF on allows for background reflected sunlight to contribute to the radiance from objects in the shade.

Another effect of using BRDF is that all of the shadows are not uniform. There is a certain amount of variability created as a result of variations in the background shape factor surrounding each point in the scene. This effect can also be seen in the real image in Figure 3.5.

Turning this option on will increase rendering time greatly. This image takes approximately 2 weeks to complete with full BRDF turned on as compared to about 12 hours without it. Both times were calculated based on Sun Blade 1000 systems with 1 gigabyte of memory. For qualitative similarity, a small amount of transmission can be added to the otherwise opaque material of the camouflage netting. This will allow more light to impact areas in the camouflage generated shade and simulate the additional radiance seen in the shade in truth imagery. This shortcut will not recreate the shape factor variability or general additional realism provided by using BRDF calculations.

Modeling the appropriate sensor effects can be accomplished once the 3-D geometry of the scene is complete. That is the focus of the next section.

3.2.4 Sensor Location and Configurations

Once the models were built and then placed in the scene at the appropriate location, the next step was to image them with virtual versions of the MISI and WASP sensors. Both sensors were located in the real scene at exactly 100 ft from the door handle of the shed approximately 45 degrees west of north. They were then lifted 50 feet by the scissor cart. The actual elevation used in the simulation is 54 ft because of the height of the aperture above the deck of the scissor cart. They are positioned in the virtual scene at the same relative coordinates.

Once the sensor locations were set appropriately, two configuration files were created to account for the two types of sensors. The visible Terrapix camera from the WASP sensor was modeled first because of the simplicity of the framing array design. It was much easier to verify the spatial creation of the MicroScene area when it was rendered in this way because none of the geometric distortions associated with MISI's line scanner design corrupted the imagery. Since the WASP sensor was only used to quickly verify relative spatial locations of objects in the scene, a scaled down version of the 16 megapixel focal plane was used to speed up the rendering time.

The MISI configuration file required a great deal more setup. DIRSIG contains routines to model sensor motion in many different ways, but simulating an airplane hovering stationary at 54 feet while pitching through a 90 degree arc was not one of them. A flight profile file needed to be created that dictated the exact position of the sensor for each individual line scan. The orientation of the sensor is described in terms of yaw, roll and pitch in DIRSIG as if it is being flown in an airplane. DIRSIG also needs to know which direction the sensor is heading in so it knows the basis for the yaw, roll and pitch. In the MicroScene configuration, MISI is stationary, so DIRSIG cannot base it usual calculation which is based on average flight direction. Therefore, the USE_SCENE_PLATFORM_ANGLES option in the configuration file was set to TRUE, so that the sensor heading was assumed to be in the positive y direction.

One of the benefits of a line scanning system like MISI is that the rendering can be broken into many pieces that can be modeled in parallel on different systems. Rendering the entire 1600 x 800 pixel image with 28 bands required nearly a week to render on a single machine, so the DIRSIG sensor profile file was broken up into 10 pieces. Each piece was rendered in about a half a day and then each piece was stitched back together using ENVI's *mosaicking* tool. This cannot be done with framing array sensors in

DIRSIG as of version 3.

3.2.5 MISI Image Calibration

Before the truth and synthetic imagery could be compared or appropriate noise could be added to the synthetic imagery, the real MISI images needed to be calibrated. The concept behind the process was described in Section 2.2.1, but a few comments will be made here to describe the actual process in ENVI for calibrating the RAW imagery.

Raw MISI data files are grouped by system board and then by bands. The bands are not necessarily in the correct order. An in-house ENVI module is available that sorts the raw MISI data into the appropriate groupings and band orders. The most important output from that process for this research is the uncalibrated 70-band, spectrometer image.

To calibrate the image, the dark region at the top of the image was selected with the ENVI's ROI tool. The mean value statistics were computed on the ROI, which produced a bias value in each of the 70 bands. The result was saved as a text file. The gain values had already been established during the calibration of the sensor and were also recorded in a text file. An important part of this step is that each of the values in the bias file needed to be negated so that when they are applied as the offset they were subtracted rather than added to the imagery. Finally, The gain and bias values were applied to the imagery by editing the header file for the uncalibrated image file in ENVI and importing the text files for both the gains and biases. To calibrate the data, the *Apply Gain and Offset* button was selected under the *Basic tools* $\Rightarrow$ *Preprocessing* menu. The gain and offset values are read directly from the image header file and the result is the calibrated image file. The same result can be achieved by using ENVI's *Spectral Math* tool.

This dark region subtraction method for removing the bias from the truth imagery was somewhat difficult due to some unknown sources of error. For example, when the GML classification was accomplished for the first time, the dark bands on each side of the 1000 truth imagery classified into different classes. This obviously pointed to some significant problem with the bias level as they both should have contained the same values. The shutter was closed during both measurements. The difference between the two regions amounted to approximately 7 radiance counts on average across all bands. At the same time, the DIRSIG and truth spectral comparisons were producing results where the truth spectra from nearly all materials was slightly higher. When the truth imagery was recalibrated with the biases from the other dark region, then the spectral curves practically fell on top of each other as will be seen in the next chapter.

The 1800 imagery did not contain a similar difference in the dark regions. Although the radiance curve comparison did produce nearly the same level of bias that was seen in the original incorrectly calibrated 1000 truth imagery. After analyzing the 1800 in depth and discussing the issue with the resident MISI experts, it was theorized that there may be an additional source of stray light that is coming through only when the shutter is open. In an attempt to determine the amount of the additional light, the grass region in the DIRSIG imagery was used as a control material because of its consistent accuracy throughout this research. A bias was then computed, based on MISI's gain values and the mean radiance of the synthetic grass, that would force the grass in the truth imagery to equal its simulated spectra. The bias levels produced a similar 7 radiance counts of difference from the original calibration. This results lends credibility to the calibration adjustment based on the same difference seen in the 1000 imagery. The truth imagery was recalibrated with the new biases. The recalibration of the 1800 truth imagery removed its consistent bias from the spectral curve comparison.

During the MISI calibration process, any band of data that is not working properly or is too far out of tolerance will be labeled as a "bad band". This bad bands list can also be imported into the images header file so that the spectral bands that cannot be relied upon are not used in any calculations. Once the bad bands were established the result was a 28 band calibrated visible MISI image and a 15 band calibrated NIR MISI image. These two images were derived from the 1000 and 1800 images respectively because of the stray light issue mentioned in Section 3.1. They are the ones used to validate the DIRSIG imagery. They are also the basis for providing information about the optical PSF of MISI as well as the amount of noise it produces. The derivation of those two phenomena will be described in the following two sections.

3.2.6 MISI Point Spread Function

Version 3 of DIRSIG does not allow the user to do PSFs on the fly, at least not completely. Basically, the PSF in DIRSIG is more of a weighted material mixer because no information is distributed between pixels. The oversampled rays cast for each pixel only affect that specific pixel. Therefore, any optical PSF must be applied in post-processing.

Before post-processing could be accomplished, MISI's PSF had to be calculated. The method based on the edge spread function described in Section 2.2.1.2 was used to extract MISI's PSF since there were insufficient resources to do a full MTF measurement in the lab. The edge with the most contrast and most consistency on each side of the transition in the truth imagery was the transition region between the sky and one of the buildings in the background shown in Figure 3.20.

The values across a single band in this region was extracted and then imported into IDL. The discrete derivative of the transition was taken to produce the LSF of the



Figure 3.20: Transition region used to determine MISI LSF

system. Then, a gaussian function was curve fit to the discrete LSF. IDL provides a function called **GAUSS_FIT** to do just this. One of the outputs of the process is the standard deviation, σ, of the fitted gaussian. The σ value returned was 0.864910. This number describes the 1-D width, in pixels, of the LSF.

MISI is a line scanner and therefore it was necessary to determine how the LSF changes in the horizontal and vertical direction. A horizontal transition region was chosen in the same location as the vertical. The σ value in the horizontal direction was 0.871707. For simplicity, the PSF was assumed to be rotationally symmetric because this result was not significantly different enough from the value of σ extracted from the vertical slice. To create the two dimensional PSF kernel, a 7x7 array was created who's values were generated from the 2-D gaussian function, which represents the rotationally symmetric LSF, described in Equation 3.1. The resulting kernel is shown in Figure 3.21.

$$2DGuass(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{3.1}$$

$$\begin{pmatrix} 0.000001 & 0.000037 & 0.000267 & 0.000515 & 0.000267 & 0.000037 & 0.000001 \\ 0.000039 & 0.001048 & 0.007544 & 0.014567 & 0.007544 & 0.001048 & 0.000039 \\ 0.000290 & 0.007783 & 0.056032 & 0.108194 & 0.056032 & 0.007783 & 0.000290 \\ 0.000566 & 0.015185 & 0.109323 & 0.211095 & 0.109323 & 0.015185 & 0.000566 \\ 0.000290 & 0.007783 & 0.056032 & 0.108194 & 0.056032 & 0.007783 & 0.000290 \\ 0.000039 & 0.001048 & 0.007544 & 0.014567 & 0.007544 & 0.001048 & 0.000039 \\ 0.000001 & 0.000037 & 0.000267 & 0.000515 & 0.000267 & 0.000037 & 0.000001 \end{pmatrix}$$

Figure 3.21: MISI PSF kernel

The PSF shown in Figure 3.21 was saved to an ASCII text file and then imported into ENVI using the *Convolutions and Morphology* tool. The PSF kernel was applied to the synthetic imagery to simulate the effects of MISI's optics on the truth imagery. Once the imagery is blurred appropriately, the synthetic noise must be added as the final step in creating a synthetic version of the MISI sensor.

3.2.7 MISI Synthetic Noise Generation

Preliminary analysis of the DIRSIG generated imagery with the RX algorithm showed that the data did not have enough variability. The multiple matrix inversions in the algorithm were constantly producing inaccurate results because of improperly formed matrices derived from regions in the image that were too stable. Therefore, it was decided to add the noise characteristic of the MISI imagery to the DIRSIG renderings. Section 2.2.1 discussed how the dark regions in the MISI imagery could be used to obtain the noise statistics of the sensor. ENVI allows detailed reports to be made that show the standard deviation of the noise in each band. The simplest solution to adding noise to the DIRSIG imagery would have been to generate random additive noise on a band by band basis with a zero mean (because DIRSIG is being rendered to match the calibrated MISI imagery) and a standard deviation equal to that of the MISI band

being rendered based on the ENVI report. Unfortunately, this solution would eliminate all of the known data about how the noise is correlated from band to band.

The technique for creating the correlated noise cubes is fairly straightforward and can be accomplished entirely in ENVI. The first step is to use the *Region of Interest (ROI)* tool to select the dark region from a MISI image and then subset that region into its own image. A rectangular ROI works best. The next step is to use the *Forward PC Rotation* button under the *Transform* menu option. From here select *Compute New Statistics and Rotate*. A menu will appear that prompts for the name of a statistics file to save the transform covariance matrix in. This file will be needed for the inverse transform later on. Then, proceed with the forward transform. The purpose of using Principle Component Analysis (PCA) is to orthogonalize the noise, which makes it completely de-correlated. The last step in this part of the process is to use the *Compute Statistics* button from the *Basic Tools* menu option. Compute the statistics of the de-correlated noise image.

Now that the forward transform statistics are known and saved, an uncorrelated noise cube can be generated using an IDL procedure developed by the author. This very simple procedure centered around IDL's **RANDOMN** function to generate random noise images with the same dimensions as the DIRSIG imagery. 28 noise images were generated to match the 28 "good" visible bands of the 1000 MISI image. The same was done for the 15 bands in the 1800 image. The key to making this work correctly is that the random numbers generated by **RANDOMN** must be multiplied by the respective standard deviation of each band computed from the forward PC transformed noise imagery. They must also be laid out in the same order, which goes from largest to smallest standard deviation because of the way the eigenvectors are sorted. This ensures that when the synthetic noise cube is back transformed, its band by band

statistics will match the original statistics. The bands are stacked and the whole cube is saved out as a single .TIF file.

Once the uncorrelated noise cube is generated with the same statistics as the forward PC transformed noise it must be imported into ENVI for the final step. This time, the *Inverse PC Rotation* button is selected from the *Transform* menu option. The synthetic noise cube is selected as the input file and then the statistics file that was generated during the forward PC rotation is used as the statistics file for the inverse. The result is a correlated noise cube with statistics that match the characteristics of the sensor and it's of the same size as the DIRSIG rendered imagery. Then, all that needs to be done is to add the noise cube to the DIRSIG imagery using the *Spectral Math* tool from the *Basic Tools* menu option. Figure 3.22 shows a graphical representation of this process.

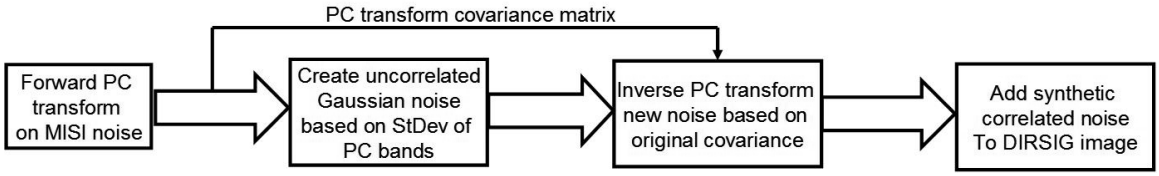


Figure 3.22: Process flow for creating correlated noise for DIRSIG imagery.

The previous sections described the unique processing steps of creating the synthetic scene. For a beginner building a similar model, a great deal of time will be spent simply learning how DIRSIG and CAD modeling tools work. It is extremely important to keep in mind the capabilities of the systems that will render the scene. A balancing act must be done between producing highly detailed objects while keeping the number of facets to a minimum. That way, memory constraints and run times will not become prohibitive. Most modeling packages allow the user to minimize the number of facets required for each object without significantly impacting the quality of the model. Trial and error

approaches will probably be required to choose the right amount of optimization.

There is obviously more work that can be accomplished to characterize how a given sensor affects the imagery it produces, but the methods described above are required to get the model accurate to at least first order. A baseline estimation of the optical PSF added to the synthetic imagery provides an added level of realism. Also, adding noise allows for algorithms to work properly by inducing local variability to the pristine surfaces generated by DIRSIG.

The last section of this chapter describes the details of the RX algorithm that needed to be programmed as an ENVI module for this research. The GLCM and GML classifier are already included, so their actual implementation will not be discussed.

3.2.8 RX Algorithm Implementation

A substantial amount of time was devoted to the implementation of the RX algorithm. The RX code used for the comparisons in this study was written in IDL and then a frontend was implemented for ENVI to facilitate its use. The ENVI frontend allowed easy implementation and distribution to other interested parties.

Figure 3.23 shows the window progression that a user would go through to run the RX algorithm. The algorithm is setup to run on any image file that can be loaded into ENVI. As demonstrated by the "RX Input File" graphical user interface (GUI), this implementation works with the same file interaction GUI's that ENVI users are familiar with.

Once a file is selected, the "RX Parameters" box appears. All of the options presented in (Reed et al., 1989) are present here. The top section of the dialog box asks the user to set two options. The first is whether to use image-wide statistics or not. Remember from Section 2.3.4, the RX algorithm primarily works by comparing the

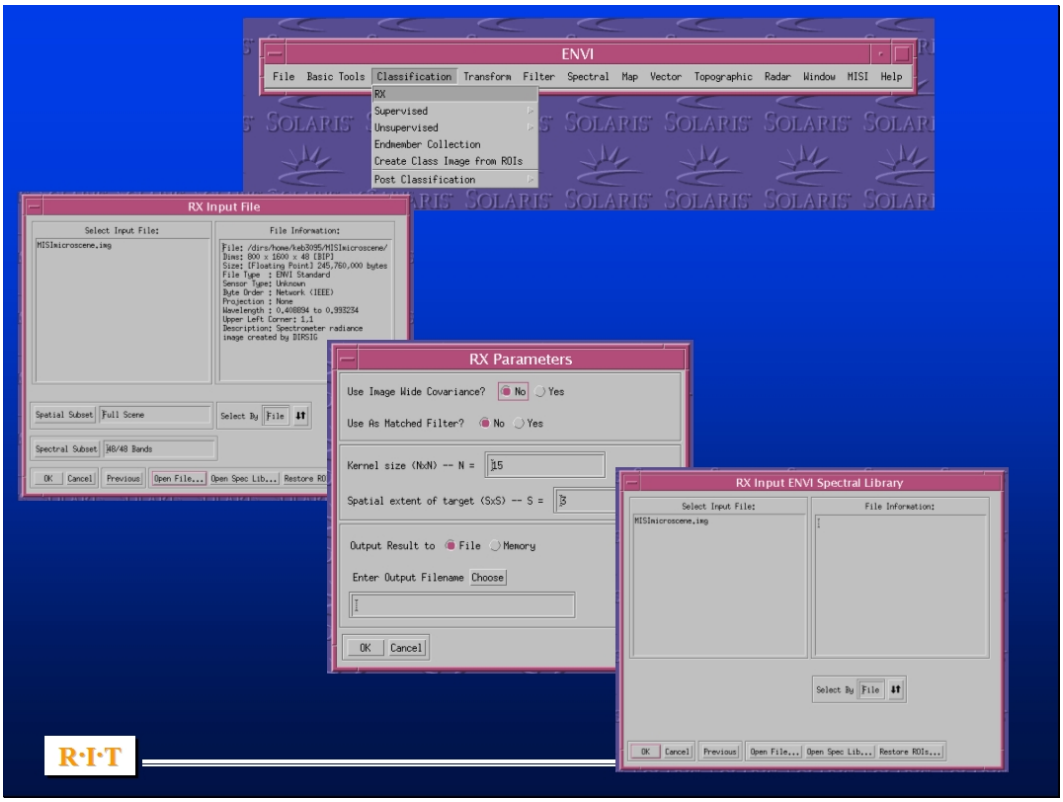


Figure 3.23: RX GUI windows.

spectral and spatial characteristics of a group of pixels to their surrounding pixels in a local neighborhood. This approach can be modified by computing the spectral covariance of the entire image and using that as the basis for the comparison. This approach reduces the ability of the algorithm to work effectively, but it works much faster since only one matrix inversion needs to be accomplished for the entire process.

The other option allows the user would like to implement RX as a matched filter. If 'No' is selected, then the algorithm works in anomaly detection mode. If 'Yes' is selected then then the user is prompted with the "RX Input ENVI Spectral Library" dialog box to input a desired target spectra in ENVI Spectral Library format. in (Reed et al., 1989), the authors run RX in anomaly detection mode and then select the spectra

$$\begin{pmatrix} N & N & N & N & N \\ N & S & S & S & N \\ N & S & S & S & N \\ N & S & S & S & N \\ N & N & N & N & N \end{pmatrix}$$

Figure 3.24: Neighborhood and spatial shape of RX kernel

of the pixel that provides the greatest return. Then, RX is run in matched filter mode with the selected spectra input as the target spectra. Users of this interface can easily recreate this approach by using ENVI's 'z-profile' tool over a pixel of interest and then saving the result as an ENVI Spectral Library. Once that is accomplished, that library file should be selected in the "RX Input ENVI Spectral Library" dialog box.

The middle section in the "RX Parameters" box asks the user to define the size of the local neighborhood to compare the target pixel against and also for the target window size. After running the algorithm several times, the author found it worked best with a neighborhood that was 45x45 pixels in area. Since the spatial shape of an unknown object is very rarely known, this interface simplifies the creation of the spatial structure of the target by only allowing symmetric differences in size to be input. The author found that a spatial shape of 3x3 or 5x5 worked best. Image noise will generate false alarms when a single pixel target windows was used. To help illustrate how this works, Figure 3.24 shows what the kernel would look like if "Kernel Size (NxN)" were set to 5x5 and the "Spatial Extent of Target (SxS)" were set to 3x3. The current implementation of the algorithm does not allow the user to define anything other than square target shapes. The reader should note that the N and S used here are not meant to correspond to those variables used in Section 2.3.4.

Finally, the bottom section of the "RX Parameters" dialog box lets the user output

the resulting RX "image" to memory or save it to a file. The *Interactive Stretching* tool under the *Enhance* menu option of an image window is the best tool for examining images produced from the algorithm. The more positive the values are, the more responsive that pixel location is, so the *Interactive Stretching* can be used to threshold the image to find the most responsive locations.

Chapter 4

Results

This chapter will present the DIRSIG validation results that were obtained at the conclusion of this research. The beginning is a qualitative analysis of the actual imagery from both MISI and DIRSIG as well as a similar analysis of the radiance spectra of the more important materials and objects in the scene.

The qualitative comparisons were by far the most useful tools for making decisions about how to update the model in the early portion of its development cycle. The spatial qualities of any scene are the first noticed by the observer and require the most focus to obtain a realistic comparison. Spectral comparisons were also extensively used throughout the development. A great deal of information can be gathered from doing simple side-by-side comparisons of radiance curves. The general accuracy of the emissivity files, weather parameters, and bi-directional effects can all be seen without the need for detailed metric analysis. The side-by-side spatial and spectral characteristics, along with the ROC results, of the scene will be presented at the beginning of this chapter.

More quantitative analysis is more important as the model matures. For most of

the metrics used here, the model needed to be generally accurate by the qualitative standards mentioned in the last paragraph before they would produce any kind of similarity in the results. The more quantitative comparisons described in Chapter 2, GML classification, GLCM texture metrics, and RX target detection will conclude this chapter. Their analysis consists of side-by-side comparisons of how they operated on both the truth and synthetic imagery. It was not initially expected that these analysis tools would provide much useful information on how to proceed with enhancing the model in future development cycles, but they were surprisingly useful. This is especially true of the GML classification and RX detection results. More on that topic will be presented in their respective sections, but first, the side-by-side imagery comparison will be presented.

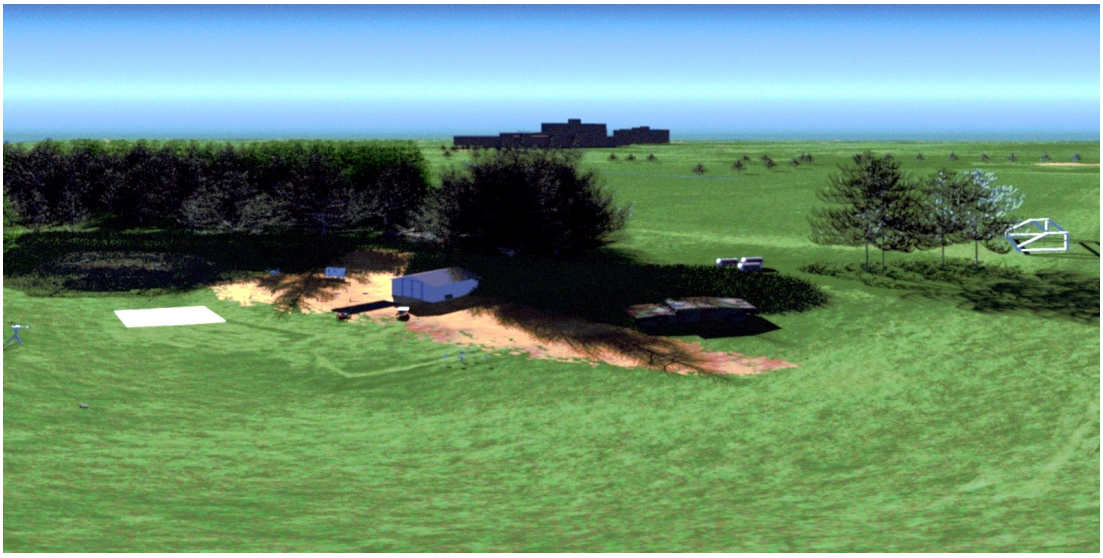
4.1 Qualitative Image Comparison

Figures 4.1 and 4.2 show the truth image on the top and DIRSIG image on the bottom for the 1000 and 1800 acquisition times. The spatial quality of the scenes are very similar. Remember, everything in the scene is empirically derived. The locations of all of the objects were taken from overhead imagery or were measured during the collection. MISI's focal length, duty cycle, PSF, noise, and detector properties were all put into the model as is, and the resulting synthetic imagery is very encouraging. The imagery shows that DIRSIG is capable of capturing the spatial distortions and tangential effects of very unique sensor configurations like the one used in this experiment.

The foreground of the DIRSIG image does a pretty good job of representing the realism of the truth imagery at this high resolution and oblique view angle. All of the terrain, vegetation, and large and small clutter in the scene produce a realistic synthetic image. The transition regions and shadowing effects also correlate well. Most



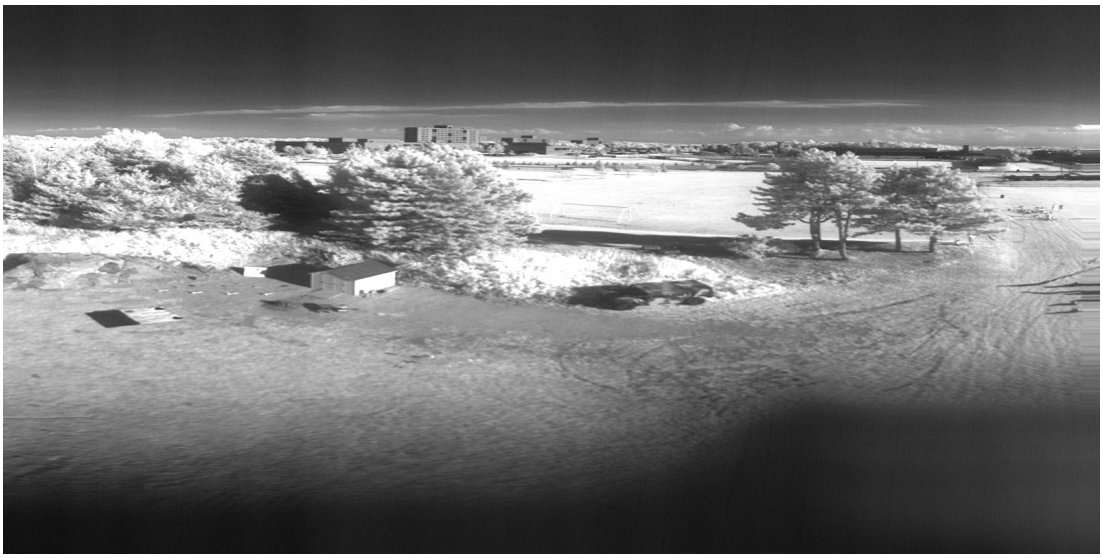
(a) MISI - 1000



(b) DIRSIG - 1000

Figure 4.1: True color MISI vs. DIRSIG image comparison.

of the scenery in the DIRSIG imagery is exactly the same in both images, but a few of the smaller targets were either moved, removed, or added between the time when



(a) MISI - 1800



(b) DIRSIG - 1800

Figure 4.2: .811 μm band MISI vs. DIRSIG image comparison.

the two images were taken. So, there are some minor differences between the two shots besides just the atmospheric and solar positioning changes associated with the time of day difference.

The most encouraging aspect of comparing the truth and synthetic imagery is that the primary targets, the Humvees, are modeled extremely well. In the DIRSIG image, all three levels of concealment produce very similar results and look very much like their truth counterparts. The time spent creating the virtual camouflage netting seems to have paid off and the Humvee in the trees shows up in nearly exactly the same manner as the real Humvee placed there.

The issues that has been found with modeling high resolution scenes at this kind of oblique viewing angle can be seen in the background of the DIRSIG imagery. The first issue is really one of time spent modeling. A literally endless amount of time can be spent making fine adjustments to the foreground portion of the scene and it only covers a small area in the total FOV of the MISI sensor. The total background region covers an area on the order of square kilometers. Unfortunately, there simply was not enough time to model the background to a similar level of detail as the foreground. The resolution requirements are not as great, fortunately, but even so, modeling realistic background variability could required as much or even more time and resources.

The second issue deals with the material and texture maps used to create the terrain variability. DIRSIG tiles the maps, described in Section 3.2.1, when the object that the map is applied to is larger than the map. For example, the dirt region, that is meant to only be in front of the shed, gets repeated off in the distance. This occurs because the material map used for the terrain needed to be extremely large to create the necessary detail in the grass-to-dirt transition regions, but it would have been too large, and consumed too many system resources, if it would have been created to fit the entire terrain map at this resolution. To the eye, this issue is barely noticeable, but it will come back up later when the anomaly detection algorithm is run on the image.

The solution didn't make it into the validation imagery, but has been updated in

the model. To eliminate the problem, the portion of the terrain model that is in the background was separated from the foreground in Rhino CAD. The foreground was left unchanged and the background material ID was reassigned from the material map ID to simply the grass ID. This eliminated the material map from being applied to this region. The texture map is still tiled though, but that cannot be seen in this imagery due to the resolution of the sensor off in the distance. Anyone who uses this model for overhead images may encounter the texture tiling in that portion of the scene.

Another less noticeable issue is that the terrain model seems to break down slightly in the far right portion of the DIRSIG imagery. The shadowing of the terrain facets contains some very jagged corners that is not realistic. This is most noticeable in the 1800 image because of the sun angle. Two things could be done to alleviate this. The first and more costly option would be increase the size of the terrain model. Since this portion of the model is not significant to the primary area of MicroScene, it doesn't make much sense. An alternative would be to change the way that DIRSIG models object shadowing. Currently, shadows are applied to objects according to the normals of their facets. A gradient based shadowing technique would be more appropriate for maintaining realistic terrain surfaces without having to increase the size of the model.

It should also be pointed out that the blurry portion at the bottom of the MISI imagery is a result of the line scanner going out of focus as the scan mirror moves the field of view closer to the sensor. Modeling this effect was not a goal of the research and so the lower quarter of the MISI imagery should be disregarded for all of the analysis in this paper.

Remember, the scene was created using a cyclical development approach. The spatial qualities of the scene were the primary focus of the first development cycle. The focus of the next portion of the model development was on the spectral properties of

objects in the scene.

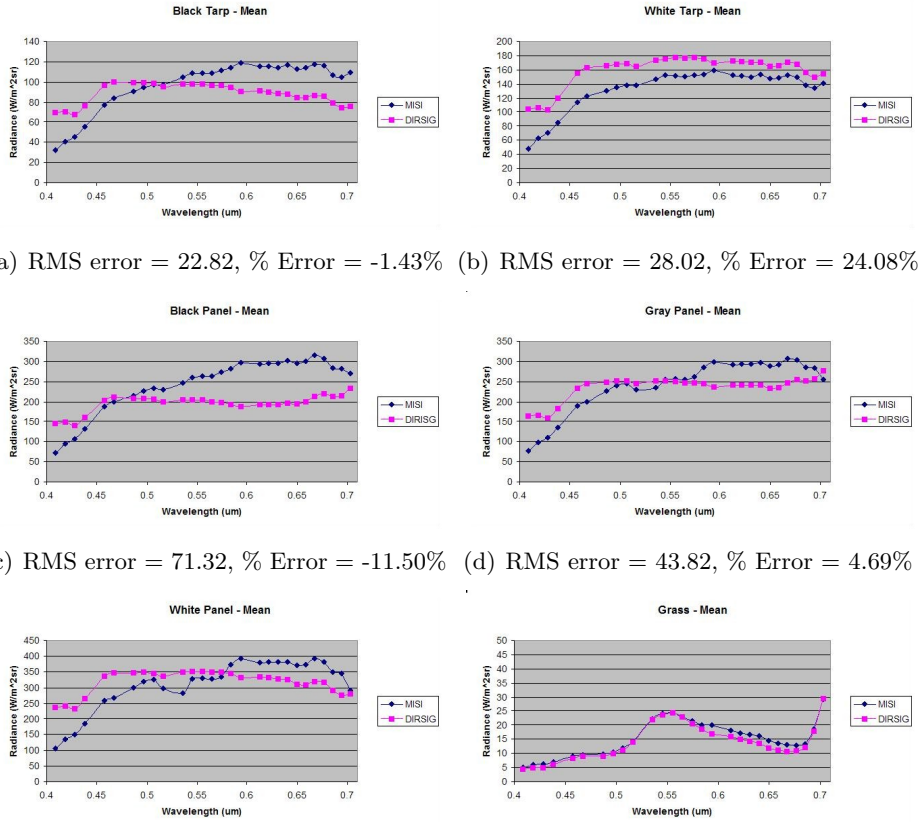
4.2 Qualitative Spectral Comparison

The spectral radiance accuracy for the visible region of the spectrum is shown in Figures 4.3 and 4.4. Radiance spectra from the NIR region is shown in Figures 4.5 and 4.6. The captions to each of the subfigures show two metrics of accuracy. The first is RMS error between the two radiance curves and the second is a percent error measurement between the DIRSIG radiance as compared to the truth spectra. The percent error is an average error accross all of the spectral bands of the given material type. The spectra in the figure were obtained by averaging the pixels in a region of interest over each of the targets.

15 targets were selected from the 1000 imagery for analysis, while only 12 were used from the 1800 imagery. In the 1800 imagery, the rollers were mostly obscured by shadows from the grassy region on the hillside. Unfortunately, the Humvee in the trees was unable to be decerned in either the truth or DIRSIG imagery due to the shadows of the surrounding trees. Finally, the portable weather station, was removed from MISI's FOV when the 1800 truth imagery was taken and so it was also removed in the synthetic shot at that time.

4.2.1 Control Target Analysis

The first five material types in Figures 4.3 and 4.5 show the accuracy of the control tarps and panels that were used in the scene. Ironically and also unfortunately, they were the most difficult material types in the scene to simulate spectrally. One of the lessons learned for obtaining ground truth imagery is that control panels should ideally be spectrally flat, and as Lambertian as possible to simplify their measurement and

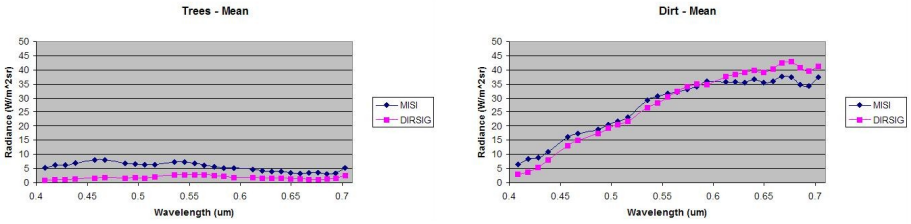


(a) RMS error = 22.82, % Error = -1.43% (b) RMS error = 28.02, % Error = 24.08%

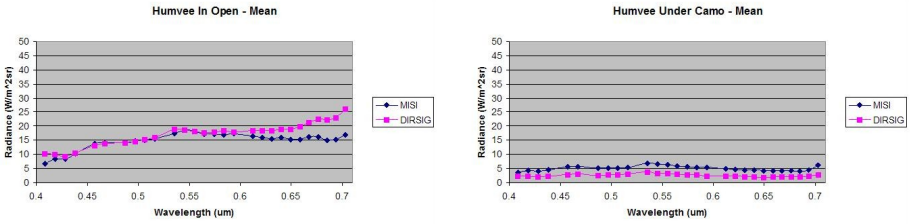
(c) RMS error = 71.32, % Error = -11.50% (d) RMS error = 43.82, % Error = 4.69%

(e) RMS error = 61.67, % Error = 9.55% (f) RMS error = 1.52, % Error = -9.20%

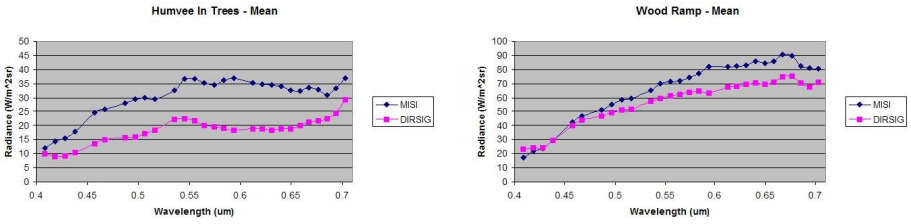
Figure 4.3: MISI vs. DIRSIG spectral comparison in the visible region at 1000.



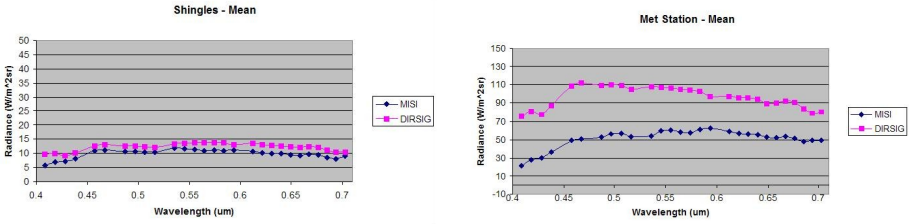
(a) RMS error = 3.92, % Error = -64.74% (b) RMS error = 3.20, % Error = -4.32%



(c) RMS error = 3.47, % Error = 15.92% (d) RMS error = 2.52, % Error = -48.05%

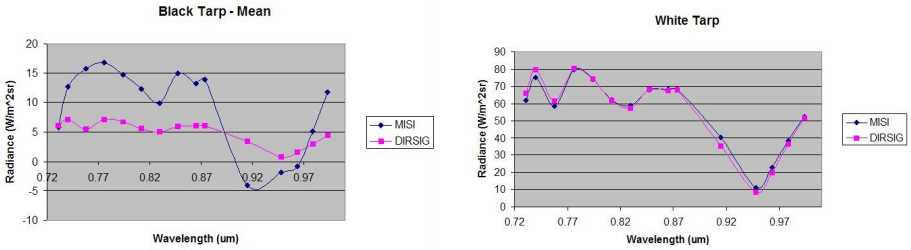


(e) RMS error = 12.49, % Error = -38.80% (f) RMS error = 10.81, % Error = -10.40%

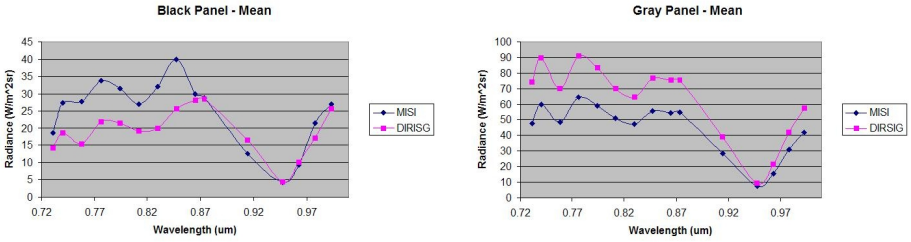


(g) RMS error = 2.47, % Error = 25.55% (h) RMS error = 46.16, %Error = 96.38%

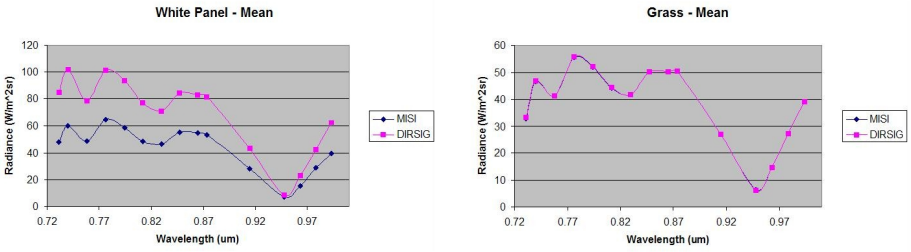
Figure 4.4: MISI vs. DIRSIG spectral comparison in the visible region at 1000.



(a) RMS error = 6.76, % Error = -49.78% (b) RMS error = 2.50, % Error = -2.48%

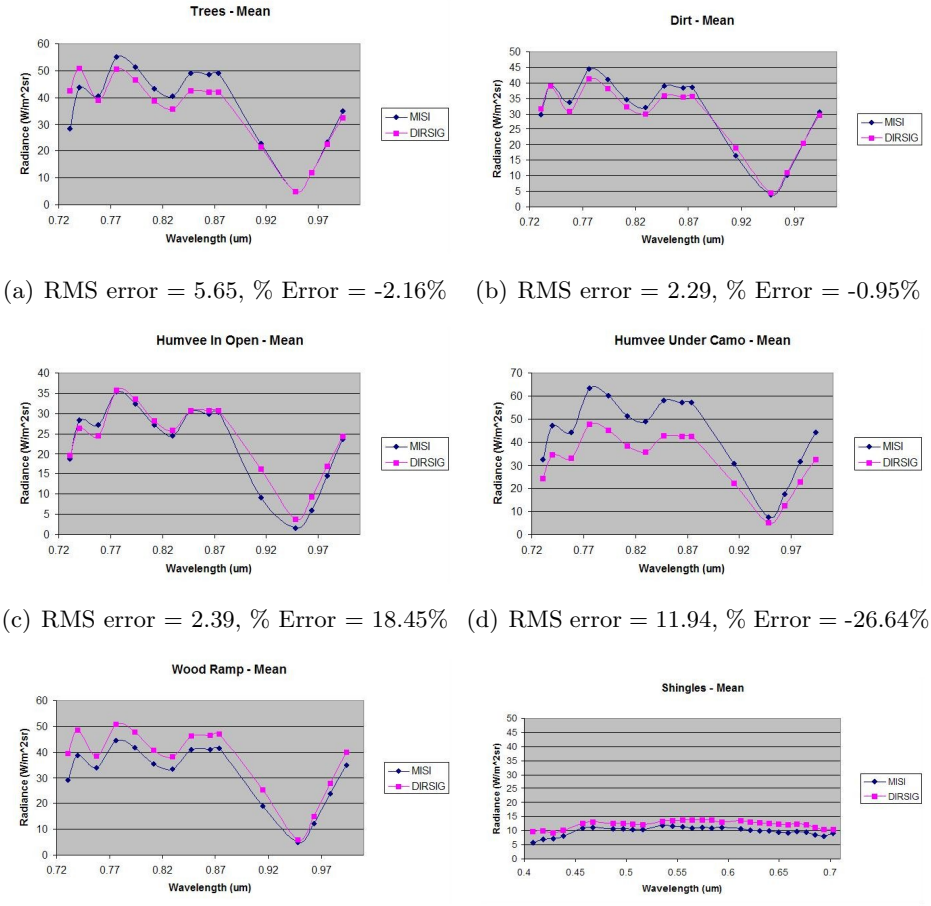


(c) RMS error = 7.88, % Error = -17.06% (d) RMS error = 19.64, % Error = 39.33%



(e) RMS error = 27.66, % Error = 54.61% (f) RMS error = 0.15, % Error = 0.03%

Figure 4.5: MISI vs. DIRSIG spectral comparison in the NIR region at 1800.



(a) RMS error = 5.65, % Error = -2.16% (b) RMS error = 2.29, % Error = -0.95%

(c) RMS error = 2.39, % Error = 18.45% (d) RMS error = 11.94, % Error = -26.64%

(e) RMS error = 5.96, % Error = 18.57% (f) RMS error = 4.11, % Error = 36.18%

Figure 4.6: MISI vs. DIRSIG spectral comparison in the NIR region at 1800.

maximize their utility as control objects. The panels and tarps that were used in the ground truth collection were selected simply because they appeared black, gray and white to the human eye and the specular properties of the materials was not given significant thought. As a result, the black tarp and all three ceramic panels initially produced simulated radiance curves that were much lower than the truth suggested in the 1000 imagery where the sun-target-sensor angle was almost exactly specular.

Because these objects are significantly non-Lambertian, DIRSIG's BRDF option was turned on for rendering the scenes. As with all of the objects in the scene, the emissivity database was constrained to only use emissivities that had been measured either in the field with the ASD instrument or in the lab with the CARY 500. Unfortunately, a measured value for the specularity parameter was unable to be accurately estimated. The materials were measured on the CARY 500 with and without the light trap inside the integrating sphere. It turns out that the size of the specular lobe of the materials was much greater than the size of the light trap, so the difference between the two measurements was not enough to produce the results seen in the truth imagery. Therefore, the specularity parameter was manually adjusted until the simulated radiance curve was as close as possible to the curves obtained from the truth imagery. The spectral plots in Figure 4.5 show the results after the specularity parameter had been adjusted.

4.2.2 Vegetation and Terrain Analysis

Obtaining accurate spectral signatures for background clutter requires a great deal of time spent focusing on variability and positioning as described in the last chapter. The most accurate spectra throughout the development of the model was the radiance measured from the grass region in the scene. As can be seen in both the 1000 and 1800

comparisons. The dirt region was also very accurate. The trees on the other hand were very difficult to model accurately.

Many of the difficulties associated with the trees was described in Section 3.2.2.3. From the beginning, the mean radiance in the trees was below their radiance of the truth imagery. Different tree models were used, different spatial configurations, the transmissive properties and also the specularity were adjusted in an attempt to make them more realistic. The problem with modeling the trees in this scene results from a phenomena that the author refers to as "tree halo". The trees in the truth imagery get brighter as one moves from their center toward the end of their branches. Also, in the truth imagery, the radiance from the trees is greater as one moves from the bottom to the tops.

Unfortunately, the author was unable to recreate this effect in DIRSIG to the same degree as can be seen in the the truth imagery. The halo effect was unable to be recreated at all with trees modeled at the branch level. It can be somewhat reproduced with trees modeled at the pine needle level, but not to the same degree of magnitude. Limited success was achieved in the 1000 imagery when the thickness of the tree facets was changed to an artificially small value. This effectively increased the transmission of the pine trees and as a result the radiance went up. Unfortunately, this is unrealistic and it also destroyed the 1800 imagery where the radiance from the trees grew by orders of magnitude. Since this experiment was conducted, DIRSIG was modified to no longer allow a violation of Kirchoff's law and emissivities and transmissions are capped so that they cannot add to a value greater than one.

Another aspect of the 1800 imagery should also be pointed out here before the classification results are presented. While the grass and trees compare favorably between the truth and synthetic imagery at this time, the truth trees and grass in the

foreground produce almost identical spectral radiance curves. This class inseparability complicates the classification of the 1800 imagery that will be presented shortly.

The grass in the background is also much brighter than the grass in the foreground in the 1800 imagery. This could be a result of the fact that this region is on one of the schools soccer fields which is watered daily. The higher radiance counts could have resulted from either having just been watered or simply that this grass is more healthy than the grass in the foreground. The spectra of the grass on the soccer field was not measured at the time and therefore was not included in the material database used in the model.

4.2.3 Humvee and Small Target Analysis

The Humvees and small targets in the scene produce radiance curves that are for the most part very similar to the truth. This is especially true of the Humvee in the open and the one under the camouflage net. The Humvee in the trees shows a similar shape, but lower general radiance count, than the truth imagery. It is possible that the specular sun-target-sensor angle is also the cause of this discrepancy. Specularity was not added to the Humvee model. Interestingly, though, DIRSIG's Humvee in the trees does show a difference of 4-5 radiance counts in the green region over the Humvee in the open. This could possibly be a result of the surrounding vegetation and show that DIRSIG is capable of rendering the effect commonly referred to as "tree shine".

The rest of the scene contains a large amount of small target clutter. Some of it contains actual measured spectra and other objects were placed in the scene simply to provide more realistic spatial variability. The emissivity of the weather station, rollers, wood ramp, and shed shingles were all measured. Each of these objects is also fairly accurate with the exception of the weather station. Only two pixels could be measured

of this target from the truth imagery because of its small size and MTF effects. Exact location, pixel phasing, and PSF measurement inaccuracy could all account for the difference. These concepts should be considered when attempting to model a similar scene at high resolution. There is a lot of potential for the model to break down radiometrically when the target is small as compared to the PSF kernel size.

4.2.4 Rank-Order Correlation Results

ROC will be presented here as another metric that the reader may be able to obtain additional validation information from. Although it is generally a more effective metric when it is used to compare images in the thermal region of the spectrum. Figures 4.7 through Figure 4.12. The results were obtained from three different spectral bands in both image sets.

Material Identification Table	
1	Humvee Under Camo
2	Trees
3	Grass
4	Shingles
5	Humvee In Open
6	Roller Tops
7	Dirt
8	Humvee In Trees
9	Wood Ramp
10	Roller Ends
11	Met Station
12	Black Tarp
13	White Tarp
14	Black Panel
15	Gray Panel
16	White Panel

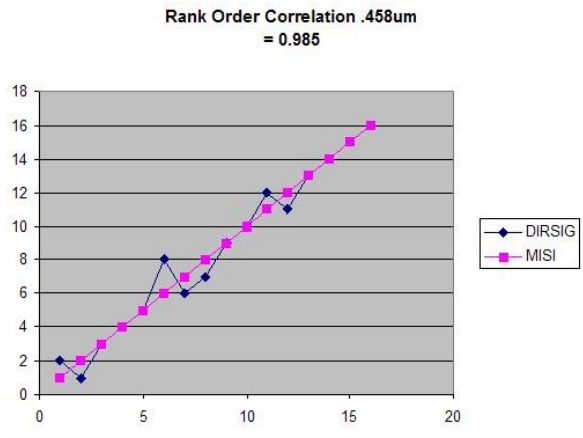


Figure 4.7: ROC results for .458μm band.

That concludes the qualitative comparison of the imagery. The spectral and spatial

Material Identification Table	
1	Humvee Under Camo
2	Trees
3	Shingles
4	Humvee In Open
5	Roller Tops
6	Grass
7	Dirt
8	Humvee In Trees
9	Roller Ends
10	Met Station
11	Wood Ramp
12	Black Tarp
13	White Tarp
14	Gray Panel
15	Black Panel
16	White Panel

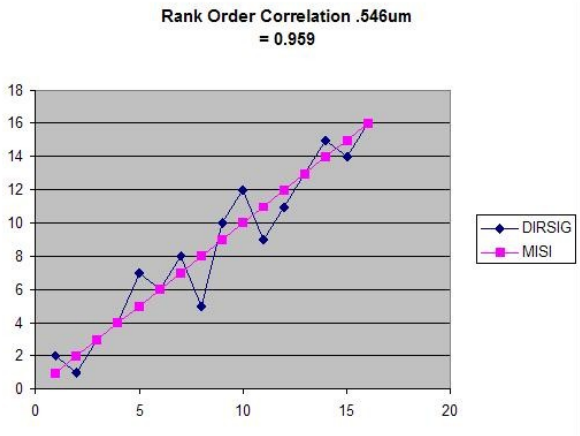


Figure 4.8: ROC results for .546μm band.

Material Identification Table	
1	Trees
2	Humvee Under Camo
3	Shingles
4	Humvee In Open
5	Grass
6	Roller Tops
7	Humvee In Trees
8	Dirt
9	Roller Ends
10	Met Station
11	Wood Ramp
12	Black Tarp
13	White Tarp
14	Gray Panel
15	Black Panel
16	White Panel

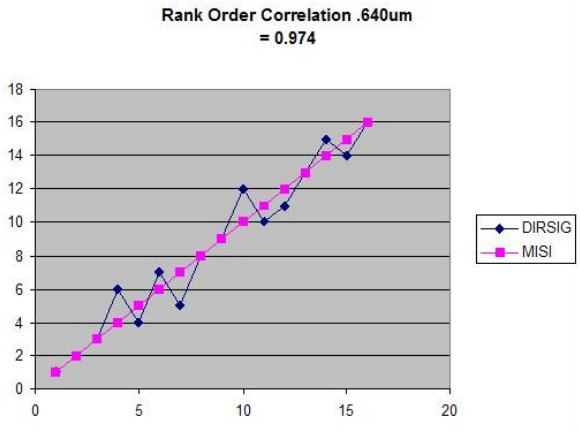


Figure 4.9: ROC results for .640μm band.

Material Identification Table	
1	Black Tarp
2	Black Panel
3	Humvee In Open
4	Shingles
5	Trees
6	Wood Ramp
7	Dirt
8	Humvee Under Camo
9	Grass
10	Gray Panel
11	White Panel
12	White Tarp

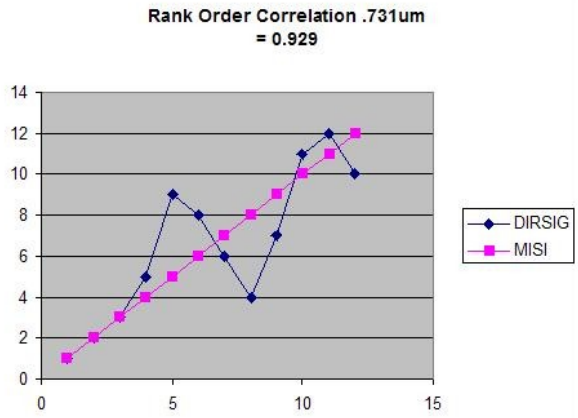


Figure 4.10: ROC results for .731μm band.

Material Identification Table	
1	Black Tarp
2	Shingles
3	Humvee In Open
4	Black Panel
5	Dirt
6	Wood Ramp
7	Trees
8	Grass
9	Gray Panel
10	White Panel
11	Humvee Under Camo
12	White Tarp

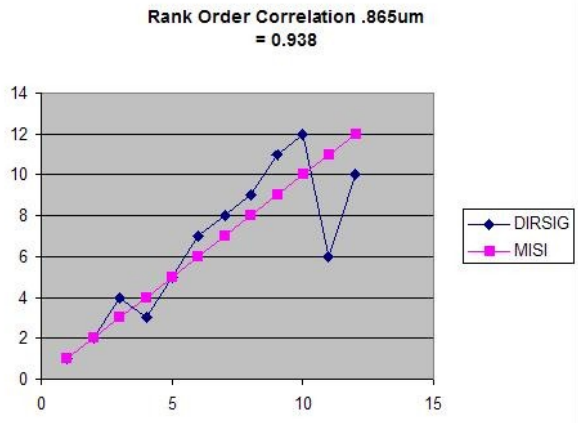


Figure 4.11: ROC results for .865μm band.

Material Identification Table	
1	Black Tarp
2	White Tarp
3	Trees
4	Humvee In Open
5	Wood Ramp
6	Grass
7	Gray Panel
8	White Panel
9	Dirt
10	Black Panel
11	Humvee Under Camo
12	Shingles

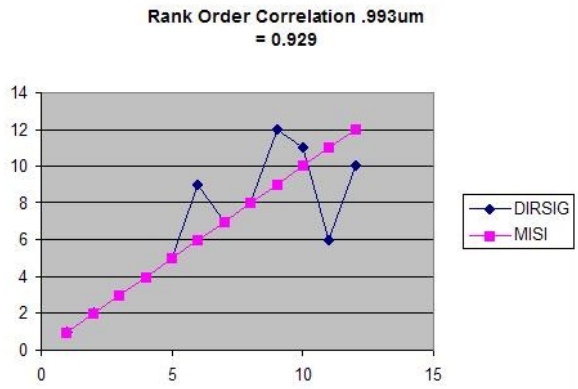


Figure 4.12: ROC results for .993μm band.

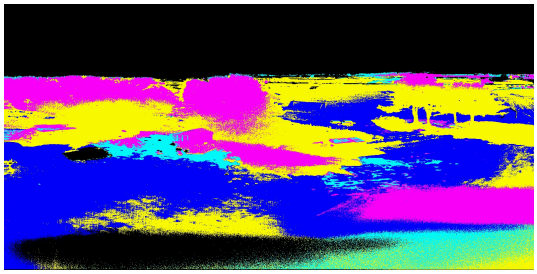
accuracy of the DIRSIG imagery is generally very encouraging. There are many more positive comparisons than negative throughout the model. This is especially true of the nearly Lambertian materials in the scene. The control panels were difficult, but with the inclusion of the simple BRDF model, even they could be forced to simulate nearly accurate radiance spectra. Some of the vegetation did not model quite right either, but their results are still fairly close. This issue is predominantly focused on the tops of the trees, where many different attempts were made to correct the issue through both spectral and spatial changes with limited success. Certain other materials in the scene were not able to be validated due to the resolution or measurement capability, such as the bark on the trees and the textured vinyl siding on the shed. For some of the small targets, generic emissivity values were used because the object was primarily in place for visual clutter and not for accurate spectral comparison modeling. One representative example of this is the red paint on parts of the generator. Generic red paint was used because of the difficulty of measuring the spectra of the generator with the tools available at RIT. In summary, all of the spectral curves used in the

comparison were measured directly from the objects being compared. It is estimated that approximately 85% of the scene is modeled with spectra that was measured directly from the objects in the scene. The rest is given representative generic spectra where required by measurement limitations or when inconsequential to the analysis. The next step in the development cycle was to see if the model was capable of training and developing classification and detection algorithms.

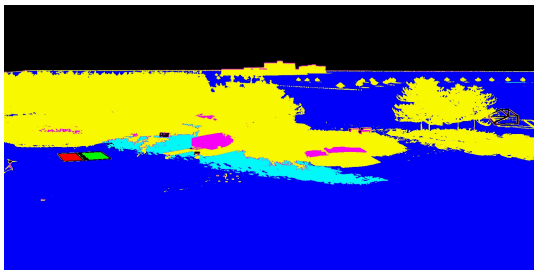
4.3 Gaussian Maximum Likelihood Classification Results

After significant data collection, scene development, and sensor characterization, the model was ready to be tested for its primary purpose of training and developing detection algorithms. Detection algorithms generally look for small targets, but the modeler cannot focus solely on the accuracy of those few pixels that encompass the target; the entire synthetic image must be accurate. It is very important that a diverse background surrounds that target so that the algorithm will have sufficient variability in the synthetic image it is operating on to perform as it would on real data. Therefore, the GML classification algorithm was selected to determine if data derived from the MicroScene model could be used to populate a training set for classifying the truth imagery. Six target classes, the dirt, grass, trees, white tarp, black tarp, and Humvee were chosen from the DIRSIG imagery. The Humvee class was taken from the top of the Humvee in the open. The resulting classification image derived from the 1000 imagery is shown in Figure 4.13(a). For comparison, the same training set was also used to classify the DIRSIG image that it was derived from. The results of that classification are shown in Figure 4.13(b).

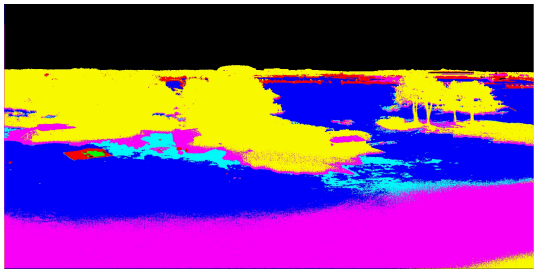
The MISI imagery still displays more variability than the DIRSIG imagery in those two figures, but the classes line up very well between the two data sets with the ex-



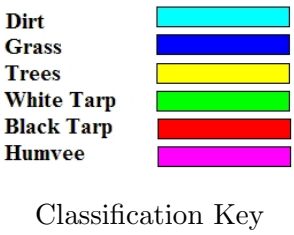
(a) GML classified MISI image using DIRSIG derived training data.



(b) GML classified DIRSIG image using DIRSIG derived training data.



(c) GML classified MISI image using MISI derived training data.



Classification Key

Figure 4.13: 1000 MISI vs. DIRSIG GML classification comparison.

ception of a great deal of Humvee class showing up in the trees. Other than that the classifier found the rest of the trees, the portion of the dirt that was in direct sunlight, and most of the grass. Portions of the Humvee are classified correctly, but it is not surprising that the Humvee was also classified as dirt and trees, since the Humvees in the scene had recently been through training exercises and had not been washed. Aside from the trees, the Humvee class also shows up in the grass, in the shadows, and around the camouflage netting. This is not all that surprising either, since the Humvee camouflage paint is meant to blend in with these objects. The shed also classifies as Humvee in both data sets.

Also of note are how the algorithm reacted to the smaller targets in the scene. For instance, unclassified portions of the imagery correlate well in both images. The most obvious region is the sky, which was not an input class, but also many of the small targets in the scene are not classified. The classifier leaving the small targets out of the general classification suggests that the anomaly detection algorithms will be able to exploit those targets. Where the small targets are classified, such as the small set of flags in the center of the image and the base of the weather station, they classify in to the same classes.

The white and black tarps are left unclassified based on the threshold confidence level that was set to produce the classification image. If the GML algorithm is set so that it classifies every pixel in the image, then a small amount of the white tarp is classified correctly in the truth imagery.

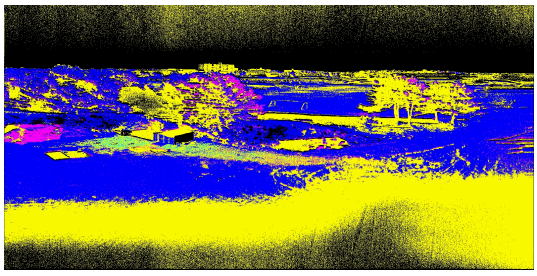
The classification image in Figure 4.13(c) was made by using training data of the same classes, but this time they were derived directly from the MISI imagery. The major difference between this image and the DIRSIG derived classification in Figure 4.13(a) is that the Humvee class is largely removed from the trees. This is a result

of the problem discussed earlier on how the trees increase in radiance as one analyzes from the bottom to the top of the trees. The radiance values produced in DIRSIG do not reach the radiance level at the top of the trees in the truth imagery and therefore they are classified into the brighter Humvee class. Overall, though, the DIRSIG derived training data does a good a job of classifying the truth imagery.

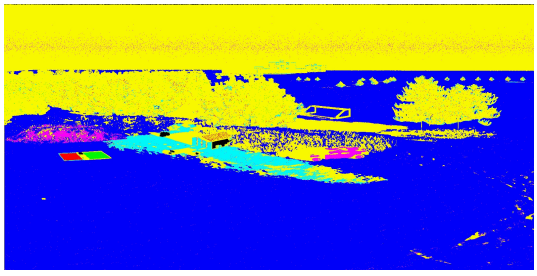
Figures 4.14(a) and 4.14(b) show a comparison of the same classification routine for the 1800 image set. The same classes were used and were obtained from the DIRSIG 1800 imagery. Unfortunately, the 1800 truth imagery is extremely difficult to classify correctly because of the lack of class separability in the truth imagery, especially between the trees and the grass. Even though the trees tend to classify as grass more than they do when the DIRSIG image is classified, the dirt and Humvee regions classify very similarly. The Humvee class shows up over the camouflage net and also in the trees where there are shadows in both images. Portions of the tarps in both image sets are classified as trees and there is even some correct classification of the white tarp. Finally, portions of the shed and background buildings both classify as trees in both images.

Again, the truth imagery was classified using truth derived classes and that result is shown in Figure 4.14(c). The image is dominated by tree class. There is also more Humvee class along the ground than in the image that was classified with the DIRSIG derived classes. This image proves the difficulty associated with classifying this scene at this time of day and sensor configuration. Even when the classes are derived directly from the image itself, the classification results are not very good.

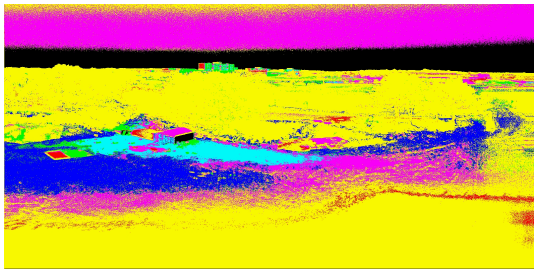
This section provided the results of general image classification using the GML classification algorithm. The results showed that DIRSIG is fairly capable of generating class statistics that can be used to classify real imagery to an acceptable level. This is



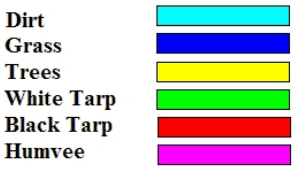
(a) GML classified MISI image using DIRSIG derived training data.



(b) GML classified DIRSIG image using DIRSIG derived training data.



(c) GML classified MISI image using MISI derived training data.



Classification Key

Figure 4.14: 1800 MISI vs. DIRSIG GML classification comparison.

very encouraging and it also proves that the background surrounding the targets in the scene have sufficient realism to allow for algorithmic detection to occur properly. One more validation technique needs to be presented before the target detection results are shown. GML classification is purely spectral in nature and gives no consideration to the spatial qualities of the scene. Since the target detection algorithm being using in this research has a spatial component, it is important to attempt to validate the spatial qualities of the scene. The next section provides the results of that validation through the use of GLCM derived texture metrics.

4.4 Grey-Level Co-occurrence Matrices Results

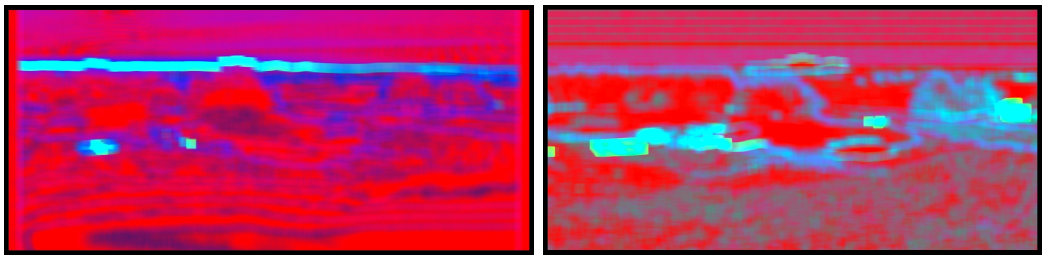
In addition to GML classification, ENVI also includes a module to compute texture images based on GLCMs. As described in Section 2.3.3, these images are difficult to interpret by themselves. Therefore, the validation of the scene using GLCM texture metrics has been broken into three pieces. The first is a simple side-by-side comparison of GLCM imagery from both DIRSIG and MISI. Next, regions of interest over the grass, trees and dirt regions in the images are compared by taking an average value across the region in an attempt to quantify the accuracy of the results. Finally, as described by Baraldi and Parmiggiani (1995), the GLCM results were added to the original truth and synthetic imagery as additional bands and then GML classification was re-performed on the new image sets.

Six different configurations were used to produce the images that are shown in Figures 4.15 through 4.18. Homogeneity, contrast, and entropy were calculated from two different GLCM runs. The first run used a one pixel shift in the x direction and a 31x31 pixel kernel. The second used a one pixel shift in the y direction with the same kernel. Homogeneity was placed in the red channel, contrast in the green and entropy

in the blue. All six configurations were run on the .555μm band of the 1000 image set and the .776μm band of the 1800 image set.

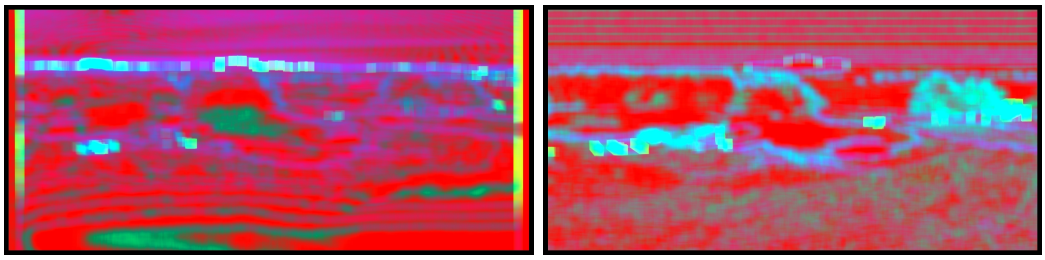
It is difficult to obtain any real answers about accuracy from these images, but in general, it appears that the "color" of the images tend to be consistent across a given material type. An interesting result of the operation is that small targets and targets with hard edges produce very unique signatures. Remember, a 31x31 kernel was used. The result seems to be that small targets, or at least small regions with high contrast edges, produce large bright blocks in the imagery. A few good examples are the soccer goal posts, the shed, and the transition from the shutter being closed for the dark scan and actual image acquisition in the MISI images where a vertical pixel shift is used.

Another observation that can be made about the GLCM images begins to reveal one of the significant issues of doing target detection or texture based image classification at a slant-angle viewing geometry. The problem lies in the spatially varying GIFOV of the sensor. While this isn't as problematic for the trees or dirt in this specific imagery, the grass texture changes significantly as one moves from the foreground into the background. The GLCM algorithm uses a fixed kernel size to produce its results and does not take this change in GSD into account. This issue will crop up again in the RX results comparison.



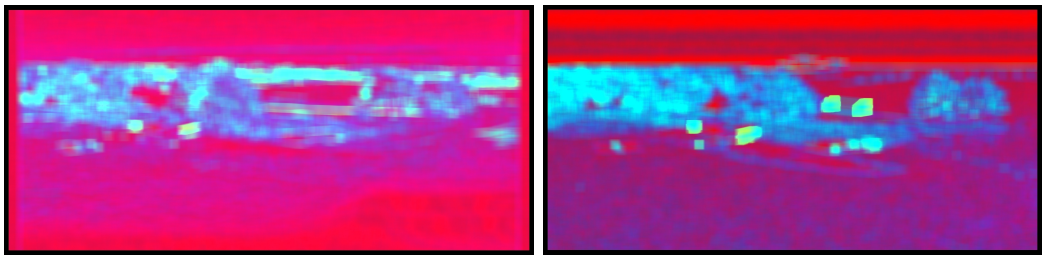
(a) MISI .555μm - horizontal GLCM. (b) DIRSIG 555μm - horizontal GLCM.

Figure 4.15: Comparison of GLCM results with a 1 pixel shift in the x direction using a 31x31 window on the .555μm band in the 1000 imagery.



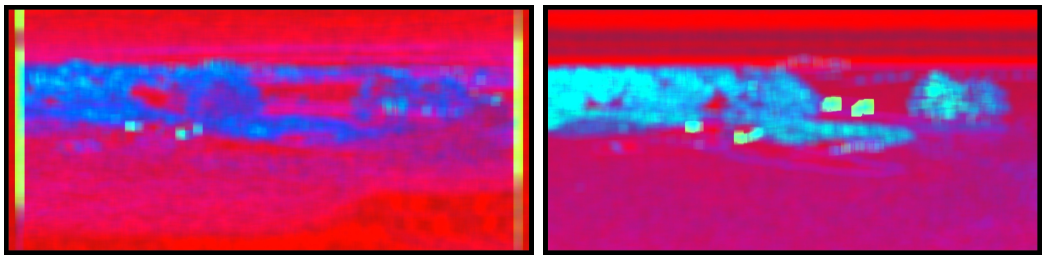
(a) MISI .555μm - vertical GLCM. (b) DIRSIG .555μm - vertical GLCM.

Figure 4.16: Comparison of GLCM results with a 1 pixel shift in the y direction using a 31x31 window on the .555μm band in the 1000 imagery.



(a) MISI .776μm - horizontal GLCM. (b) DIRSIG .776μm - horizontal GLCM.

Figure 4.17: Comparison of GLCM results with a 1 pixel shift in the x direction using a 31x31 window on the .776μm band in the 1800 imagery.



(a) MISI .776μm - vertical GLCM. (b) DIRSIG .776μm - vertical GLCM.

Figure 4.18: Comparison of GLCM results with a 1 pixel shift in the y direction using a 31x31 window on the .776μm band in the 1800 imagery.

Figures 4.19 through 4.21 are plots of mean GLCM values for homogeneity, contrast and entropy that are derived from the grass, trees and dirt from truth and synthetic

imagery. All of these plots show very encouraging results. In every one, with the exception of Figure 4.20(c), the value trends are very consistent between both data sets. It is unclear why the trees produce such a significantly greater contrast value in only the horizontally shifted GLCM image of the .776μm band.

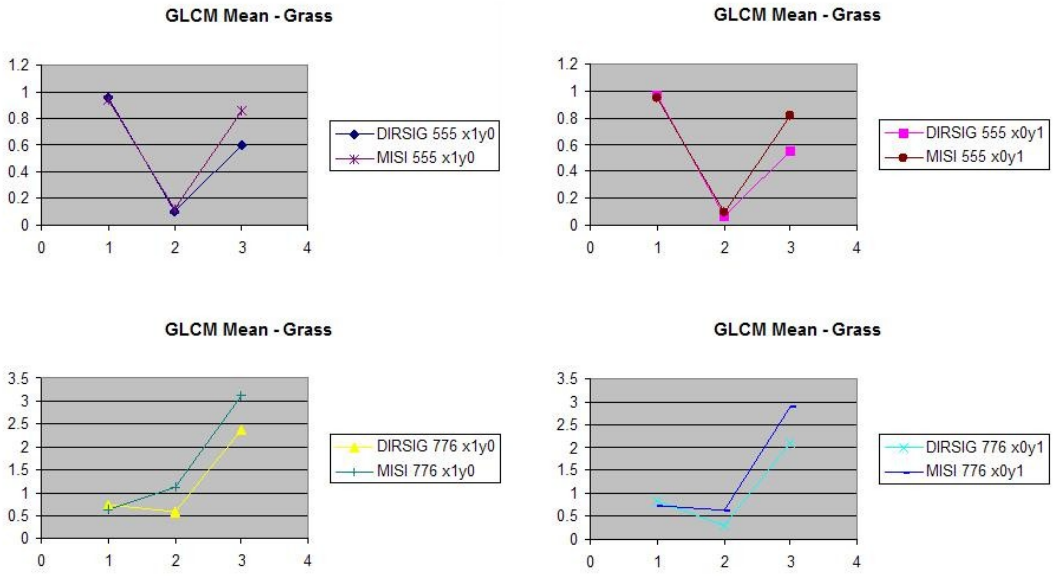


Figure 4.19: Mean GLCM values of grass regions : 1=Homogeneity, 2=Contrast, 3=Entropy.

Since the plots of the GLCM metrics seem very encouraging, each of the six GLCM configurations was added back to the image that it was derived from as additional bands of data. Then, GML classification was run on the new image. In Section 4.3 the classification algorithm strictly operated on the radiance of single pixels. Once the GLCM bands were added back to the original imagery, those new bands act as a spectral representation of the spatial texture of the surrounding neighborhood of pixels. In essence, the GML classifier is now basing its classification decisions on both spatial *and* spectral information.

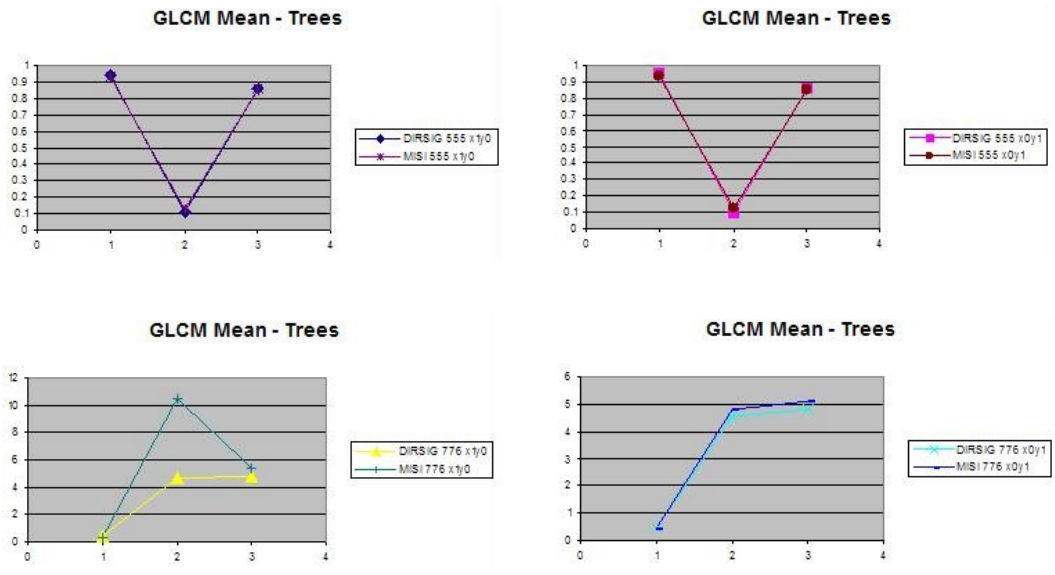


Figure 4.20: Mean GLCM values of trees regions : 1=Homogeneity, 2=Contrast, 3=Entropy.

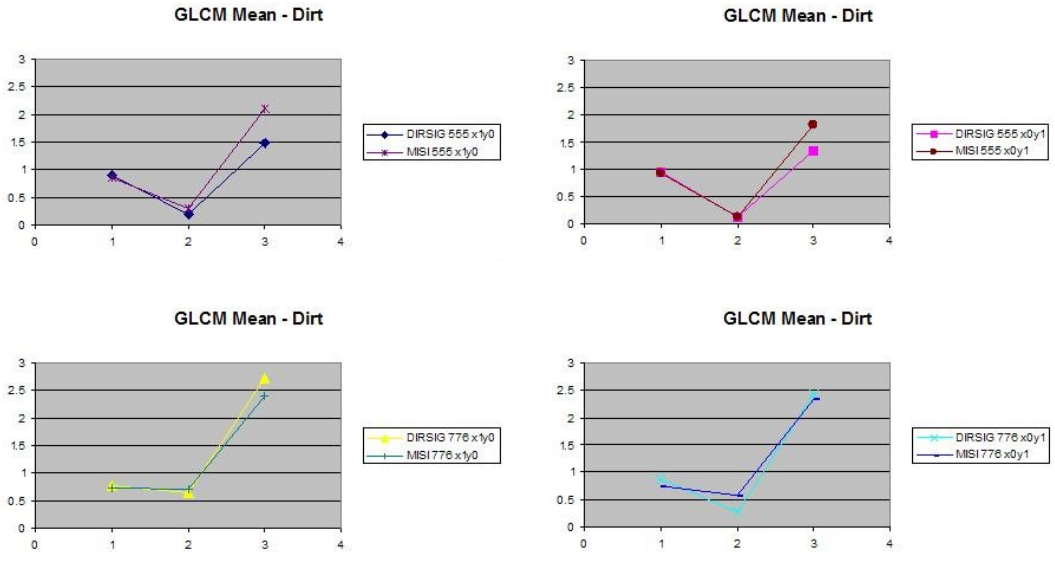


Figure 4.21: Mean GLCM values of Dirt regions : 1=Homogeneity, 2=Contrast, 3=Entropy.

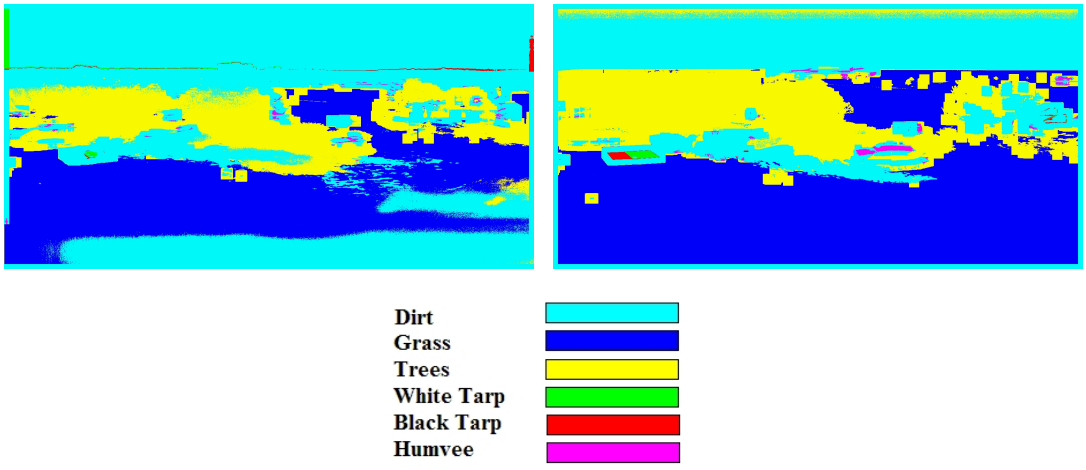
The results of running the GML classifier on the 1000 image set with the GLCM bands added back is shown in Figure 4.22. The same six classes were used as before and the similarity between the two images is very impressive. Also, compared to the original classification results (see Figure 4.13(a) and 4.13(b)) the Humvee no longer shows up in anywhere near the quantity that it did before, which significantly increases the detection rate if this were the algorithm being used to find Humvees in the imagery. The top of the trees in the truth imagery is still not classified into the correct class, but at least more of the trees are classified correctly. Also, the grass is much more accurately classified whereas before it contained a lot of stray tree class. The new results also correctly classify portions of both the black and white tarp. This was not the case originally.

The 31x31 kernel used to create the GLCM images acts as somewhat of a low-pass filter to the resulting imagery. The small target issues discussed above are a result of the kernel size. For this reason, the entire image was allowed to be classified as one of the six classes with no unclassified pixels. Keeping unclassified regions in an attempt to find anomalous small targets was not possible because the small targets were essentially destroyed by the kernel.

Also, the kernel size produces blocky classification. The bright spots around high contrast edges that can be seen in Figures 4.15 through 4.18 heavily influence the classification results in the window around that edge. The trees on the right hand side of the two image sets get classified as blocks of dirt and tree in both GLCM classification images in Figure 4.22. Some small targets, such as the flag set, get classified as blocks of trees in both image sets, as well.

This same technique was used on the 1800 image set, but the results were not successful. Essentially the entire image was classified as trees. Apparently, the GLCM

metrics only served to provide additional weight to the tree class and the entire classification of the difficult to separate spectra were overpowered by the dominant tree class.



Classification Key

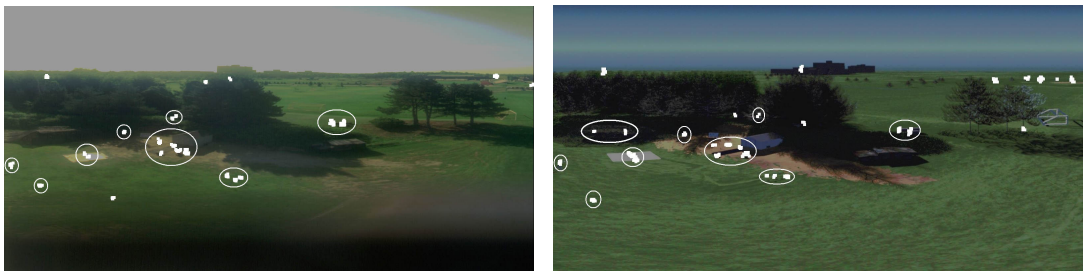
Figure 4.22: GML classification of 1000 imagery that includes the derived GLCM images.

That concludes the discussion of the general image comparison and classification. From the results presented so far, DIRSIG is capable of accurately recreating the spatial and spectral qualities of truth imagery. Also, both the spectral and spatial properties of the synthetic imagery can be used to derive class statistics with enough quality for doing reasonably accurate real image classification. The final step in this validation is to provide the results of running a target detection algorithm on the two data sets. The next, and final, section in this chapter is devoted to the RX algorithm results.

4.5 RX Results

RX performs well in regions where the targets are small, relative to the neighborhood size chosen, and are significantly different from their background. While an exhaustive analysis of the algorithm was not accomplished, it was determined that the best results were found when a 45x45 pixel neighborhood was used with a 5x5 spatial subset. The success criteria for this research is not to determine the best way to use any particular algorithm, but to generate synthetic imagery that produces similar results to real imagery under target detection analysis.

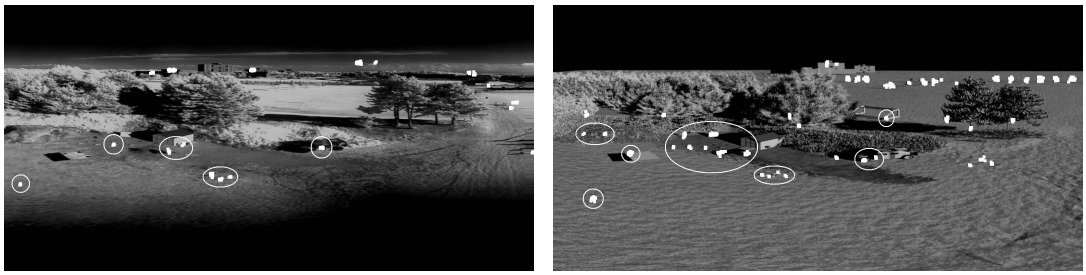
The results of running the RX algorithm in anomaly detection mode on the MISI and DIRSIG images are shown in Figure 4.23. The gray-scale image produced by the algorithm was thresholded and then the entire image was dilated with a 9x9 kernel for presentability. In Figure 4.23, the circled regions represent known targets in the scene that were detected by the algorithm. Also for presentability, the original image has been added back to the RX image to help the reader identify the targets identified by the algorithm.



(a) MISI - 1000 (b) DIRSIG - 1000

Figure 4.23: RX run in anomaly detection mode on 1000 RX imagery.

Most all of these targets were detected in both images. All other points shown in the anomaly detection images that are not circled are either false alarms or interesting



(a) MISI - 1800 (b) DIRSIG - 1800

Figure 4.24: RX run in anomaly detection mode on 1800 RX imagery.

targets that are unknown and therefore were not modeled. The results are nearly identical for the detected objects in the two scenes. The objects that are detected are the weather station, the control panels, portions of the Humvee in the open and the one in the trees, the generator and blackbody in front of the shed, the rollers, the downwelled radiometer, and finally a group of small flags and a rock. The Humvee under the camouflage was not considered anomalous in either truth images and the Humvee in the open was not detected in the 1800 truth imagery. The Humvee in the trees was not detected in either the truth or synthetic 1800 imagery as it is completely hidden in the shadows.

Originally, it was thought that only the qualitative spatial and spectral comparisons would be useful for making changes to the model, but the anomaly detection algorithm turned out to be very important as well. The group of small flags and the rock were not in the model until after the RX algorithm was run on the scene. They both produced significant returns in the truth imagery even though they were thought to be insignificant. Therefore, they were both added to the latest version of the model. This shows that these algorithms can not only validate, but also help to identify improvements and provide direction in to the next step in the continuous improvement of the model.

A comparison of the false alarms in Figure 4.23 are also fairly consistent. Portions

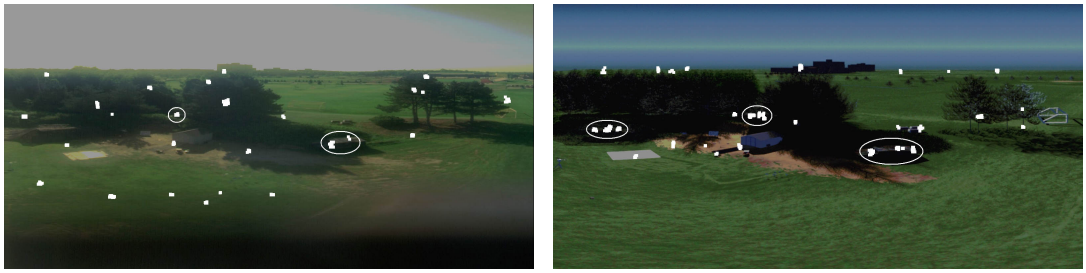
of the skyline and tops of the trees are selected as anomalies by the algorithm in both sets of images. RX also has the habit of producing seemingly random results over uniform areas. This becomes more evident when the algorithm is run in matched filter mode as will be discussed shortly.

Some of the false alarms in the DIRSIG imagery are a result of the material map tiling issue discussed earlier. The dirt region repeats in the distance. As it gets farther out, it's total size in pixels gets closer to the 5x5 square target region of the RX kernel. The algorithm only sees dirt in the target pixels and grass in the neighborhood pixels. More time spent modeling detail into the background would help decrease these artificial false alarms.

As mentioned in the section on the GLCM results, the oblique viewing angle of the scene produces spatially varying GIFOVs of the sensor. This impacts the results of the algorithm's performance. Objects that would be large, and not considered anomalous, if they were in the foreground of the image, produce large RX results in the background. The top right portion of the 1800 DIRSIG imagery in Figure 4.24 show a great deal of this issue. In addition to the tiled dirt region, there are a number of trees that appear almost exactly in the 5x5 window of the algorithm and also produce large results. This phenomenon should be considered when doing off-nadir analysis using the RX, or any other spatially dependant, algorithm. Fortunately, the DIRSIG model reproduces this effect accurately so that the RX results are similar between it and the truth imagery with respect to this issue.

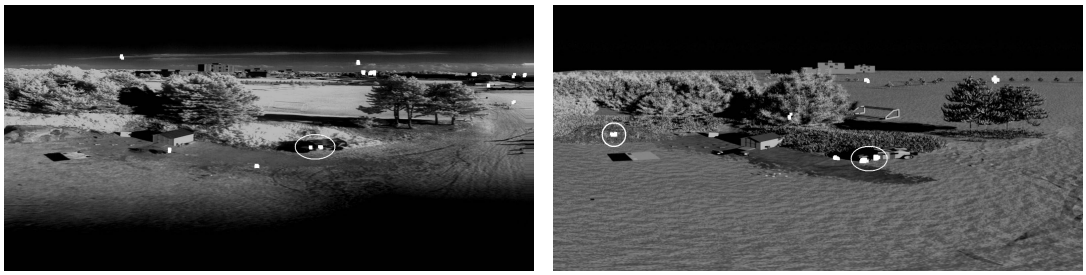
Figures 4.25 and 4.26 show the results of the RX algorithm running on both sets of data in matched filter mode. The input spectra for the algorithm was selected as the mean radiance spectra of the Humvee in the open. That seemed like a reasonable amount of a priori knowledge for running this kind of detection in the field. The

threshold point for the RX results was chosen at the point where the Humvee in the trees was just classified as a target in the truth imagery. Again, the same threshold point was used in the DIRSIG image. Target points in the images in Figures 4.25 and 4.26 are only circled if they are over the location of one of the Humvees.



(a) MISI - 1000 (b) DIRSIG - 1000

Figure 4.25: RX run in matched filter mode using DIRSIG derived Humvee on 1000 imagery.



(a) MISI - 1800 (b) DIRSIG - 1800

Figure 4.26: RX run in matched filter mode using DIRSIG derived Humvee on 1800 imagery.

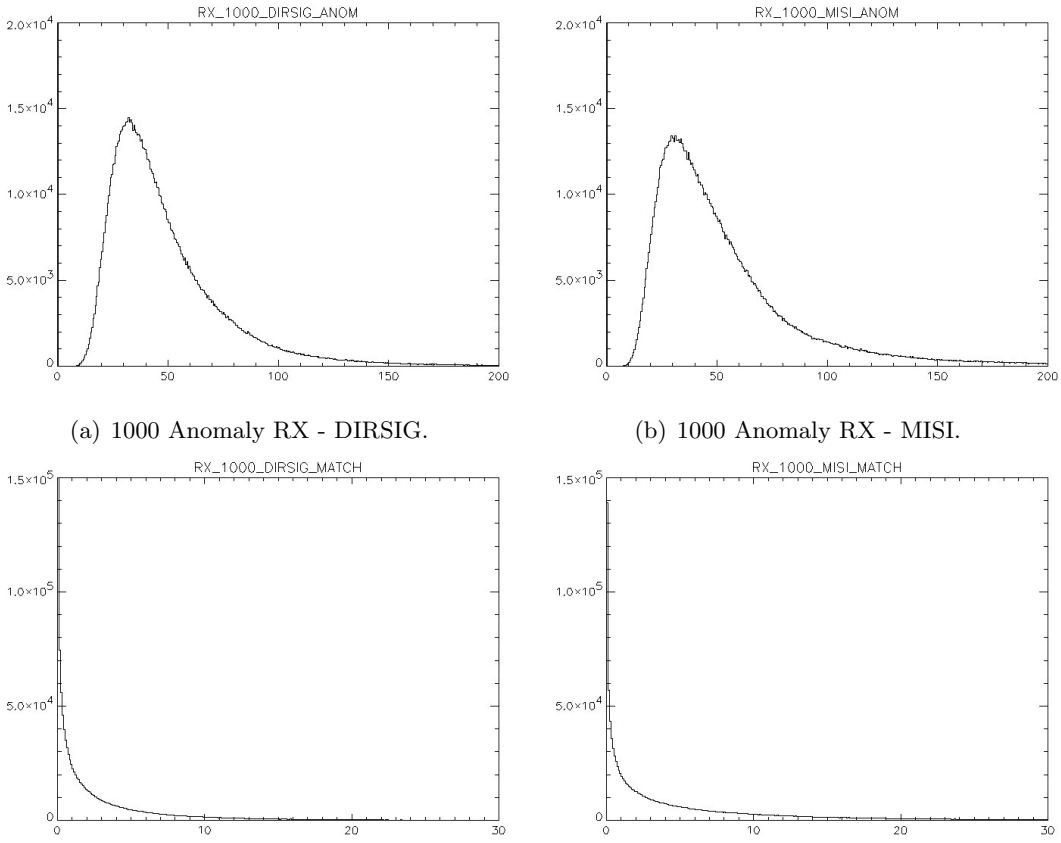
This time, the algorithm finds the Humvee in the trees and also the Humvee in the open in both 1000 images. In the DIRSIG image, the Humvee under the camouflage netting also gets detected in that data set. There are a significant number of false alarms in both image sets, especially at 1000. In general, a number of the small targets in the scene were still detected in the DIRSIG image, while they were not in the truth.

Also, the truth imagery contains a quite a few other, seemingly random, false alarms scattered across the image. Analysis of the truth imagery in these locations did not reveal any obvious reason for this. While not as prevalent in the DIRSIG image, a few of the false alarms in that image are also over regions where there is no target of interest.

Only the Humvee in the open was detected in the 1800 truth imagery. Again, the Humvee in the trees is invisible due to the shadows. There are fewer false alarms in the two images in contrast to the 1000 imagery. These false alarms are also seemingly random, with no noticeable reason for the large return value. From a comparison perspective, though, both data sets produce similar numbers of false alarms in both the 1000 and 1800 data sets. This fact alone lends credibility to the models ability to perform accurately under the algorithms scrutiny.

The histograms of the gray-scale RX images prior to thresholding are also important. The plots of those histograms can be seen in Figures 4.27 and 4.28. The histograms of the gray-scale RX images are nearly identical between the truth and synthetic images. This was the reason why the exact same threshold value used to threshold the truth imagery could be used to threshold the synthetic imagery. The similarities in the histograms show that the detection rates for the synthetic imagery will be very comparable to the truth imagery regardless of the threshold level chosen. This also lends credibility to DIRSIG's ability to generate synthetic imagery useful for target detection algorithm training.

In all, the algorithms produced very similar results. The RX algorithm appears to react to insignificant portions of the scene, especially when used in matched filter mode. Even though these regions did not always match up perfectly between the truth and synthetic imagery, they did produce similar quantities of false alarm rates in all



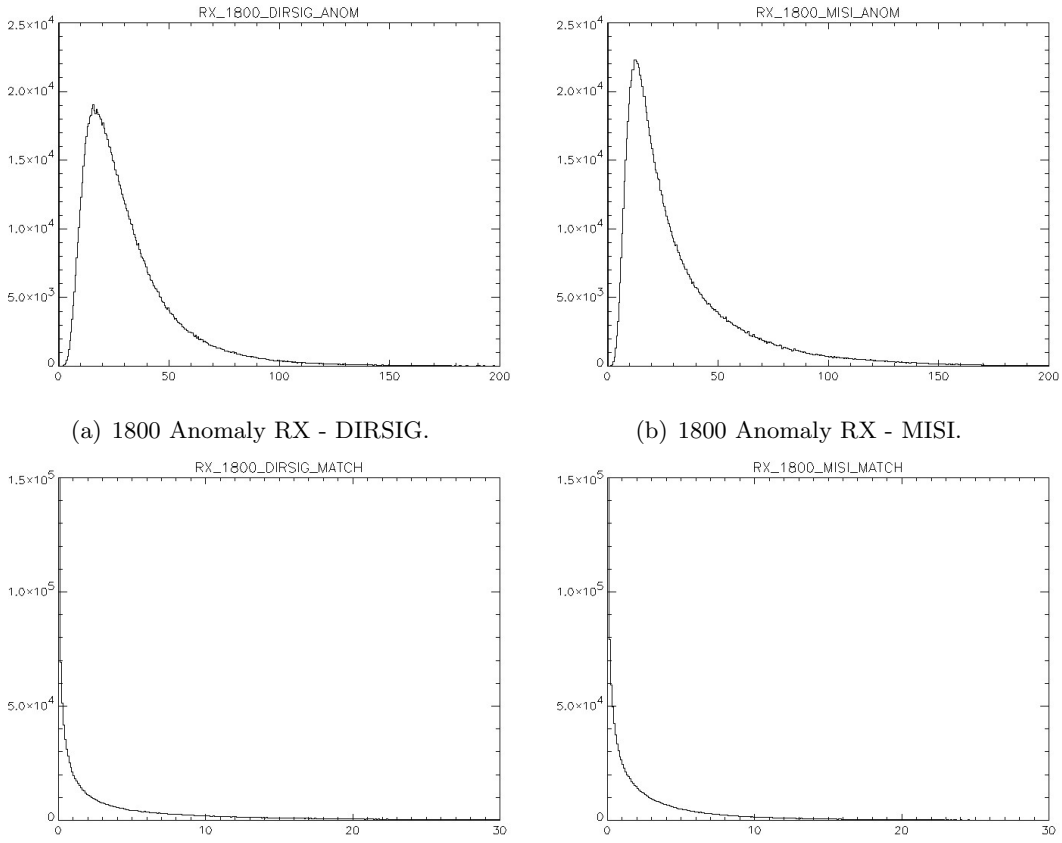
(a) 1000 Anomaly RX - DIRSIG.

(b) 1000 Anomaly RX - MISI.

(c) 1000 Match RX - DIRSIG.

(d) 1000 Match RX - MISI.

Figure 4.27: Histograms of the RX algorithm results from the 1000 imagery.



(a) 1800 Anomaly RX - DIRSIG.

(b) 1800 Anomaly RX - MISI.

(c) 1800 Match RX - DIRSIG.

(d) 1800 Match RX - MISI.

Figure 4.28: Histograms of the RX algorithm results from the 1800 imagery.

the image sets. The RX algorithm was even surprisingly discovered to be an effective means of identifying necessary improvements to the model. This occurred because the algorithm identified objects in the truth image that were not expected to be of significance and were not initially modeled.

This completes the validation of the DIRSIG model as compared to the truth imagery of the scene the model was based on. Many metrics were used to assist in detailing the ability of the DIRSIG model to reproduce real scenes accurately for the purpose of image classification and target detection. The final portion of this paper will present some conclusions recommendations for further study that have been drawn from these results.

Chapter 5

Conclusions and Recommendations

The goal of this research is to determine DIRSIG's ability to model real scenes accurately enough so that it could be used as a training tool for algorithm development. It consisted of three major parts. The first part was collecting truth imagery to be used as the basis for the comparison. The second portion was to construct a virtual replica of that scene and then render the scene in DIRSIG. Finally, the two scenes were compared using a number of analysis tools.

The ground truth collection was largely a success. An enormous amount of data was obtained that included spectral information from the visible through the long-wave portion of the spectrum using two different imaging instruments. Many people from across the Imaging Science department came together to make this happen through a great deal of planning and imagination. To the author's knowledge is was the first collection of its kind, where that type and amount of data was collected using a stationary imager setup. In addition to all of the data collected on concealed targets, there is

an additional amount of thermal information that was collected in support of buried landmine detection.

Even with all of that preparation, there were still a number of lessons learned during and after the collection was accomplished. Those issues primarily deal with the control objects that are used and the stray light that plagued most of the MISI spectrometer imagery. Some detector bias calibration issues were also addressed in this research, but the uncertainty in the MISI imagery is not enough to discredit the results of the DIRSIG comparison.

The creation of the synthetic model of the MicroScene area required considerably more time than initially expected. That additional time required to get the model to its current state took time away from diving deeper into the finer points of the scene. The trees and background are the two main areas that could still use more time. The background simply requires additional clutter to make it more appealing to the viewer, but the trees are a different story. Multiple attempts were made to recreate the radiance level seen in the truth imagery without as much success as is thought possible. Many different tree models were tried as well as geometric configurations.

A number of tools were used to accomplish this main validation objective. Qualitative comparisons between the DIRSIG and truth images were accomplished of the general spatial and spectral aspects of the scene. Also, more quantitative comparisons were made. First, ROC curves were produced. The scene was also classified using the GML classification algorithm. Next, GLCM derived texture metrics were run on the imagery. These results were combined into the original data set and GML classification was rerun on the hybrid scene. Finally, the RX target detection algorithm was run on the data sets.

Overall, the results of this research show that DIRSIG is capable of being used to

train and develop classification and target detection algorithms. Visual comparisons of the spatial and spectral qualities of the scene show that DIRSIG produces very similar results to truth imagery. These results are confirmed under different times of day and while maintaining an adherence whenever possible to actual measured material properties and sensor characteristics. DIRSIG appears to handle the unique viewing geometry and high-resolution without any significantly restrictive problems. Even the extremely high resolution shot taken from underneath the camouflage netting produced a very realistic result. That image alone validated many of DIRSIG's capabilities, such as shape factor determination and material maps with NULL material IDs. It also proved how intricate models can be created with relatively few facets if DIRSIG's hierarchical mapping structure is used appropriately.

The general truth image classification using DIRSIG derived training classes worked fairly well in the 1000 imagery where there was more class separability. The classification process was not as successful in the 1800 imagery. The results of the GLCM metrics was more successful than initially expected. Those results were powerful enough to make the classification of the 1000 imagery even more accurate. Finally, the RX algorithm showed a great deal of similarity between the truth and synthetic imagery in both anomaly and matched filter mode. Small targets in both scenes were similarly detected and false alarm rates were comparable.

A few issues were discovered along the way that another DIRSIG user trying to develop a similar scene would need to keep in mind. First, DIRSIG will tile the image maps used to induce object variability. Depending on the viewing geometry and system constraints, this may or may not become an issue. The anomaly detection results showed how this issue can produce false alarms.

Also, DIRSIG models near-Lambertian materials much more accurately than specu-

lar ones. This is a result of two issues. First, there is very limited support for modeling specularity in DIRSIG 3. Second, there is currently no means available at RIT for measuring the bi-directional reflectance properties of different material types. So, while all of the Lambertian material properties in the scene were used as they were measured from either the ASD field instrument or the CARY 500, there was no way to measure non-Lambertian properties. This issue was especially prevalent in the black control tarp and all of the control panels. The radiance values of these objects in the truth imagery were used as the basis for tuning the specularity value of those material types in the DIRSIG material file.

The final topic is the potential directions that MicroScene could be taken in the future. The first and foremost of these new directions deals with the needs of the Army MURI group. When this project was proposed, more information was supposed to be provided to the author in terms of what types of phenomenology should be validated in this model. The specifics of the term "phenomenology" were never relayed and so the analysis tools that were used here were selected as the author's idea of what the model should be able to accomplish with regard to concealed target detection. As with any simulated scene, we will never have the knowledge or computing power to be able to recreate the infinitely complex interactions that occur in the real world. That is why it is unrealistic for people to assume that any model is phenomenologically accurate without reference to some specific phenomena. Therefore, future work on this project should be accomplished based on the suggestions of the phenomenology and modeling segments of the Army MURI team to determine what specific phenomenology should be targeted for modeling accuracy. Then, whatever algorithm is determined to do the best job at detecting concealed targets, based on that phenomenon, can be used as a more appropriate metric for validating the scene.

Second, this scene is ideal for doing more detailed analysis of how DIRSIG models vegetation at high-resolution. The pine trees in the scene required unique modeling solutions that would not have been required in lower resolution scenes. But still, the differences seen between the truth and synthetic scenes with respect to the trees is enough to make one believe that there is phenomenology associated with the tiny interactions of the needles that is manifesting itself in the truth imagery that DIRSIG may or may not be able to recreate.

The spatial realism of the trees may also become and issue if further research is done with active systems, such as LIDAR or RADAR. Many applications for these types of sensors are geared toward elevation estimation. The virtual forest was planted from a 2-D overhead image. Therefore the locations of the trees in the scene is very accurate, but no detailed measurement of the height of those trees was accomplished. The actual height of the forest canopy in MicroScene will need additional scrutiny before the model can be used for accurate simulation of height variability using any of DIRSIG's more advanced active sensor models.

In addition to the trees, more three-dimensional detail will be needed if the resolution requirements increase much beyond the model's current capability of approximately 3 inches. The sub-inch resolution of the WASP camera shows individual blades of grass and more variability in the grassy region on the hill. The hierarchical 2-D mapping structure of DIRSIG that worked so well for modeling the terrain may need to be replaced by 3-D grass models. Currently, the hill is modeled three-dimensionally, but with somewhat uniform grass and stalks of weeds placed randomly throughout. The model of this region is currently limited to only the grass material ID. In reality there are different types of plants with small flowers and slightly different colored leaves. At higher resolutions, these nuances may become important. This variability was indistin-

guishable in the MISI truth imagery. Also, no 3-D plant variability is included in the model beyond the foreground. The forest floor is only modeled with the 2-D texture map and the 3-D trees. This may not be realistic for targets hidden in that area of the scene.

The model is also ideally suited for being expanded into the thermal portion of the spectrum because of the large amount of thermal information that was obtained during the truth imagery collection. Both aspects of the Army MURI project, concealed target and landmine detection, could potentially benefit from thermal properties being added to this model. To accomplish this, the sun location correction described in Section 3.2.1 would need to be mitigated in the next version of DIRSIG and many of the material emissivity files would need to be expanded beyond $2.5\mu m$. Aside from those two issues, the scene geometry and truth imagery are already in place.

Another area for future work on this model is to add empirically derived BRDF information about the materials and objects in the scene. Once the capability to measure this type of information becomes available at RIT, and DIRSIG is capable of handling it, the model should be revalidated with that more accurate spectral information.

Finally, the power of the model could really be expressed if this type of research was done in reverse. Since it has been shown that a model can be built from information gathered from a real scene, then it is the logical next step (and more realistic premise) that a synthetic scene should be created and then comparison imagery could be taken. This is truly the conditions that the ultimate user of the Army MURI group's research would be operating under in the field. In this more realistic scenario, a computer would be tasked to simulate a scene and the target conditions prior to an operator being there physically. Successfully demonstrating DIRSIG's ability to generate realistic scenes and a priori data sets under these conditions would be a significant step toward the ultimate

goal of using simulated data to train target detection algorithms.

Bibliography

A. Baraldi and F. Parmiggiani. An investigation of the textural characteristics associated with gray level coourrence matrix statistical parameters. In *IEEE Transactions on Geoscience and Remote Sensing*, volume 33, pages 293–304, March 1995.

A. Berk, G.P. Anderson, P.K. Acharya, J.H. Chetwynd, L.S. Bernstein, E.P. Shettle, M.W. Matthew, and S.M. Adler-Golden. *MODTRAN4 User's Manual*. Air Force Research Laboratory, Hanscom AFB, MA 01731-3010, 1999.

S.D. Brown. *DIRSIG Course Materials Release 3.5*. Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY 14423-5604, 2003a.

S.D. Brown. http://www.cis.rit.edu/research/dirs/dirsig/about/overview/index.html. Online, 2003b.

S.D. Brown and J.R. Schott. Characterization techniques for incorporating backgrounds into DIRSIG. Presented at the SPIE AeroSense Conference, April 2000.

J. M. Cathcart and A D. Sheffer. Generation ad application of high-resolution infrared computer imagery. In *Optical Engineering*, volume 30, pages 1745–1755, November 1991.

CIS. http://www.cis.rit.edu/research/dirs/research/misi.html. Online, 2003.

DCS Corporation. AIRSIM thermal signature prediction and analysis tool model assumptions and analytical foundations. DCS Technical Note 9090-002-001, 1991.

Department of the Army. Camouflage, concealment, and decoys. Field Manual 20-3, August 1999.

T.J. Doll. http://eoeml-web.gtri.gatech.edu/tdoll/viseo_gtsims/sld007.htm. Online, 1997.

T.J. Doll and R. Home. Guidelines for developing and validating models of visual search and target acquisition. *Optical Engineering*, 40(9):1776–1783, September 2001.

T.J. Doll, S.W. Mcwhorter, D.E. Schmieder, M.C. Hetzler, J.S. Stewart, A.A. Wasilewski, W.R. Owens, A.D. Sheffer, G.L. Galloway, and S.D. Harbert. 'VISEO' system for camouflage/LO design based on computationa vision research. In *Proceedings of the HAVE FORUM Low Observables Symposium*, 1997.

R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, second edition, 2001.

D. Filbee, A. Krik, C. Stroud, G Hutchings, T. Ward, and D. Brunnen. Modeling of high fidelity synthetic imagery for defence applications. In *Targets and Backgrounds VIII: Characterization and Representation*, volume 4718 of *Proceedings of SPIE*, pages 12–22, 2002.

J. Francis, L. Maver, and J.R. Schott. Comparison of physically and computer generated imagery. In *Image Modeling*, volume 1904 of *Proceedings of SPIE*, pages 20–33, 1993.

R. Hailstone. *Basic Principles of Imaging Science, I.* Center for Imaging Science, Rochester Institute of Technology, 2002.

R.M. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. In *IEEE Transactions On Systems, Man, and Cybernetics*, volume SMC-3, pages 610–621, November 1973.

E. Ientilucci. http://www.cis.rit.edu/research/dirs/resources/megascene_doc/c74.htm. Online, 2003.

D.J. Joseph. DIRSIG: A broadband validation & evaluation of potential for infrared imaging spectroscopy. Master's thesis, Rochester Institute of Technology, Center for Imaging Science, March 1998.

R.K. Kiang. Utilizing spatial features in classifying high-resolution imagery data. In *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery VIII*, volume 4725 of *Proceedings of SPIE*, pages 267–274, 2002.

T.A. Kraska. DIRSIG: Digital imaging and remote sensing image generation model: Infrared airborne validation and input parameter analysis. Master's thesis, Rochester Institute of Technology, Center for Imaging Science, April 1996.

J.S. Loyd and J.S. Sanders. Physically realistic camouflage net models for visualization and signature generation. volume 4370 of *Procedings of SPIE*, 72-83, 2001.

D. Manolakis. Overview of algorithms for hyperspectral target detection theory and practice. In *Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery VIII*, volume 4725 of *Proceedings of SPIE*, pages 201–215, 2002.

S.R. Marscher, H.W. Jensen, M. Cammarano, S. Worley, and P. Hanrahan. Light scattering from human hair fibers. In *ACM Transactions on Graphics*, volume 22 of *Proceedings of ACM SIGGRAPH 2003*, pages 780–791, July 2003.

J. Mason, J. Schott, C. Salvaggio, and J. Sirianni. Validation of contrast and phenomenology in the digital imaging and remote sensing (DIRS) lab's image generation (DIRSIG) model. In *Advanced Signal Processing: Algorithms, Architectures, and Implementations V*, volume 2269 of *Proceedings of SPIE*, pages 622–633, 1994.

D. McKeown. http://www.cis.rit.edu/info/ia_s2003_talks/mckeown_talk.pdf. Online, 2003.

K. Mitcell, J. Moorhead, G. Watson, M. Thomson, T. Yates, T. Troscianko, D. Tolhurst, and D. Filbee. Trial SNAPSHOT: measurements for terrain background characterisation. In *Targets and Backgrounds VI: Characterization, Visualisation and the Detection process*, volume 4029 of *Proceedings of SPIE*, pages 256–266, 2000.

K. Mitcell, J. Moorhead, G. Watson, M. Thomson, T. Yates, T. Troscianko, D. Tolhurst, and D. Filbee. Assessment of synthetic image fidelity. In *Targets and Backgrounds VIII: Characterization and Representation*, volume 4718 of *Proceedings of SPIE*, pages 23–34, 2002.

K. Mitchell, J. Moorhead, M. Gilmore, G. Watson, M. Thomson, T. Yates, T. Troscianko, and D. Tolhurst. Assessment of synthetic image fidelity. In *Targets and Backgrounds VI: Characterization, Visualization, and the Detection Process*, volume 4029 of *Proceedings of SPIE*, pages 256–266, 2000.

I. Moorhead, M. Gilmore, A. Houlbrook, D. Oxford, D. Filbee, C. Stroud, G. Hutchings, and A. Kirk. CAMEO-SIM: a physics-based broadband scene simulation tool for asssessment of camouflage, concealment, and deception methodologies. In *Society of Photo-Optical Instrumentation Engineers*, volume 40(9) of *Optical Engineering*, pages 1896–1905, September 2001.

I. Moorhead, M. Gilmore, T. Liddicoat, D. Oxford, D. Filbee, C. Stroud, G. Hutchings, and A. Kirk. CAMEO-SIM: a broad-band scene generation system that is 'fit for purpose'. In *Targets and Backgrounds: Cahracterization and Representation V*, volume 3699 of *Proceedings of SPIE*, pages 217–228, April 1999.

S. Nyberg and L. Bohman. Assessing camouflage using textural features. In *Targets and Backgrounds VII: Characterization and Representation*, volume 4370 of *Proceedings of SPIE*, pages 60–71, 2001.

I.S. Reed and X. Yu. Adaptive multiple-band CFAR detection of an optical pattern wwith unknown spectral distribution. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 38, pages 1760–1770, October 1990.

I.S. Reed, X. Yu, and A.D. Stocker. Multidimensional signal processing for electro-optical target detection. In *Signal and data processing of small targets*, volume 1305 of *Proceedings of the SPIE*, pages 218–231, 1989.

J.A. Richards. *Remote Sensing Digital Image Analysis*. Springer-Verlag, Berlin, 1999.

N. Scanlan. Implementation, validation and comparative performance analysis of texture characterization models in DIRSIG. Master's thesis, Rochester Institute of Technology, Center for Imaging Science, 2003.

J.R. Schott. *Remote Sensing: The Image Chain Approach*. Oxford University Press, Oxford, NY, 1997.

J.R. Schott, S.D. Brown, R.V. Raqueno, H.N. Gross, and G. Robinson. Advanced synthetic image generation models and its application to multi/hyperspectral algorithm development. Proceedings of the AIPR Workshop in Washington DC, October 1998.

J.R. Schott, S.D. Brown, R.V. Raqueno, H.N. Gross, and G. Robinson. An advanced synthetic image generation model and its application to multi/hyperspectral algorithm development. *Canadian Journal of Remote Sensing*, 25(2), June 1999.

J.R. Schott, S.D. Brown, and M.J. Richardson. The role of physical modeling and scene simulation in support of space based remote sensing. In *Applications of Remote Sensing I*, Core Technologies for Space Systems Conference, 2001.

A.D. Sheffer, J.M. Cathcart, and N.L. Faust. High-fidelity infrared scene simulation at georgia tech. In *High-Fidelity Simulation for Training, Test Support, Mission Rehearsal, and Civilian Applications*, volume 2740 of *Proceedings of SPIE*, pages 142–152, June 1996.

G.W. Walker and J.R. McManamey. The importance of second-order statistics for predicting target detectability. In *Characterization, Propagation, and Simulation of Sources and Backgrounds III*, volume 1967 of *Proceedings of SPIE*, pages 308–319, 1993.

R.F. Walker, P.T. Jackway, and I.D. Longstaff. Recent developments in the use of the co-occurrence matrix for texture recognition. In *Digital Signal Processing Proceedings*, volume 1 of *13th International Conference on*, pages 63–65, July 1997.

M. Wellfare, D. Vechinski, J. Watson, J. Foster, J. Edwards, and M. Richards. Irma 5.0 multi-sensor signature prediction model. In *Targets and Backgrounds VI: Characterization, Visualization, and the Detection Process*, volume 4029 of *Proceedings of SPIE*, pages 217–235, 2000.

R.A. White. Validation of rochester institute of technology's digital image and remote sensing image generation model - reflective region. Master's thesis, Rochester Institute of Technology, Center for Imaging Science, July 1996.

Appendix A

RX Code

```
pro RX_doit, fid=fid, dims=dims, pos=pos, N=N, b=b, IMAGE_WIDE=iw, S=S, $
        out_name=out_name, in_memory=in_memory

  ;N = number of pixels alone one side of kernel
  ;b = an optional spectral vector FOR matched filter version.
        ; If NOT set THEN b = X_bar##S
  ;IMAGE_WIDE = only compute a single covariance based on image wide statistics
  ;S = the spatial shape kernel

  !error=0
  on_ioerror, trouble
  in_memory = KEYWORD_SET(in_memory)

  ENVI_FILE_QUERY, fid, fname=fname, xstart=xstart, ystart=ystart
  ns = dims[2] - dims[1] + 1
  nl = dims[4] - dims[3] + 1

  orig_image = FLTARR(ns, nl, n_elements(pos))

  FOR i=0, n_elements(pos)-1 DO BEGIN
        orig_image[*,*,i] = ENVI_GET_DATA(fid=fid, dims=dims, pos=pos[i])
  ENDFOR

  sz = intarr(3)
  sz[0] = n_elements(pos)
  sz[1] = ns
  sz[2] = nl

  GET_LUN, unit
  IF (in_memory) THEN $
        mem_res = FLTARR(sz[1], sz[2]) $
  ELSE $
    OPENW, unit, out_name

  L = sz[0]       ;number of bands

  IF NOT KEYWORD_SET(b) THEN BEGIN
    b = FLTARR(1,L)
    match_filter = 0
  ENDIF ELSE match_filter = 1

  ;initialize variables
  M_inv = FLTARR(L,L)    ;inverted covariance matrix of subwindow
  r = FLTARR(sz[1],sz[2])
  avg=FLTARR(1,L)
```

```
Krnl = FLTARR(N,N)              ;spatial subwindow
Krnl[N/2-S/2:n/2+S/2,N/2-S/2:N/2+S/2] = 1.0/FLOAT(S)
S =  reform(Krnl, 1, n_elements(Krnl))

;initialize r
r[*,*] = 0

IF (iw) THEN BEGIN
  print, "Making Image Covariance"
  avg_orig_image = FLOAT(orig_image)  ;initialize Mean_X to size of X

  FOR i = 0, L-1 DO BEGIN
  avg_orig_image[*,*,i] = FLOAT((total(orig_image[*,*,i]))) $
                      /FLOAT(sz[1]*sz[2])
      ENDFOR

      X_bar = orig_image - avg_orig_image

      ;reorder X_bar into a vector
      X_bar = reform(X_bar, N_elements(X_bar[*,*,0]),L)

      ;compute covariance
      M = (1.0/FLOAT(N_elements(X_bar[*,0])))*(X_bar)##TRANSPOSE(X_bar)
      M_inv = invert(M,status,/double)
      print, 'status = ', status
ENDIF

;;;;;;;;;;;;;;;;;;;

s0=0l
s1=0l
s2=0l

;Begin Computation of r by moving NxN kernel through image
IF (in_memory) THEN tstr = 'Output to Memory' $
ELSE tstr = 'Output File : ' + out_name
ENVI_REPORT_INIT, ['Input File: ' + fname, tstr], $
      title='RX Processing', base=rbase, /interupt

      ENVI_REPORT_INC, rbases, sz[1]-1-N

      FOR i = N/2, sz[1]-1-N/2 DO BEGIN
  FOR j = N/2, sz[2]-1-N/2 DO BEGIN

            ENVI_REPORT_STAT, rbase, i, sz[1]-1-N, cancel=cancel
            IF (cancel) THEN BEGIN
          !error = envi_cancel_val()
                GOTO, trouble
            ENDIF

            IF (iw) THEN BEGIN

              ;create X from NxN subwindow from orig_image
              X = FLOAT(orig_image[(i-N/2):(i+N/2),(j-N/2):(j+N/2),*])

              X_bar = X - avg_orig_image[n_elements(X)]

              ;reorder X_bar into a vector
              X_bar = reform(X_bar, n_elements(X_bar[*,*,0]),L)

              ENDIF ELSE BEGIN

              ;create X from NxN subwindow from orig_image
              X = FLOAT(orig_image[(i-N/2):(i+N/2),(j-N/2):(j+N/2),*])
              Mean_X = X

              FOR c = 0, L-1 DO BEGIN
                    Mean_X[*,*,c] = FLOAT(total(X[*,*,c]))/FLOAT(N*N)
              ENDFOR

              X_bar = X - Mean_X

              ;reorder X_bar into a vector
              X_bar = reform(X_bar, n_elements(X_bar[*,*,0]),L)

              ;compute covariance
              M = (1.0/FLOAT(N_elements(X_bar[*,0])))*(X_bar)##TRANSPOSE(X_bar)
              M_inv = la_invert(M, status=status, /double)
```

```
                IF (status EQ 0) THEN s0=s0+1l
                IF (status EQ 1) THEN s1=s1+1l
                IF (status EQ 2) THEN s2=s2+1l

              endelse

              IF NOT match_filter THEN BEGIN
                temp_b = X_bar##S                    ;anomaly detector version
              ENDIF ELSE BEGIN
                IF iw THEN temp_b = b-avg_orig_image[0,0,*]
                IF NOT iw THEN temp_b = b-Mean_X[0,0,*]
              endelse

              ;compute r
              r[i,j] = ((TRANSPOSE(temp_b)##M_inv##X_bar##S)^2) $
                       /((1.0-1.0/FLOAT(N*N)*TRANSPOSE(X_bar##S)##M_inv##(X_bar##S)) $
                       *(TRANSPOSE(temp_b)##M_inv##temp_b))
          ENDFOR
        ENDFOR

        print, 'Good Inversions:  ', s0
        print, 'No Solution Inversions:  ', s1
        print, 'Bad Approximation Inversions:  ', s2

        IF (in_memory) THEN $
          mem_res = r $
        ELSE $
          writeu, unit, r

        !error = 0
        trouble: IF (!error NE 0) THEN $
          ENVI_IO_ERROR, 'RX Processing', unit = unit
          FREE_LUN, unit

          IF (!error EQ 0) THEN BEGIN
              descrip = 'RX Processing'
              IF (in_memory) THEN $
              ENVI_ENTER_DATA, mem_res, descrip=descrip, $
                xstart=xstart+dims[1], ystart=ystart+dims[3], $
                r_fid=r_fid $
              ELSE $
                ENVI_SETUP_HEAD, fname=out_name, ns=ns, nl=nl, nb=1, $
                data_type=4, interleave=0, xstart=xstart+dims[1], $
                ystart=ystart+dims[3], /write, /open, r_fid=r_fid, $
                descrip=descrip
              ENDIF

  ENVI_REPORT_INIT, base=rbase, /finish
end

pro RX_handler, ev

  WIDGET_CONTROL, ev.id, get_uvalue=uvalue

  IF (uvalue EQ 'rx_handler') THEN BEGIN
              ENVI_SELECT, title='RX Input File', fid=fid, dims=dims, pos=pos

        IF (fid EQ -1) THEN RETURN


        ;Build ENVI widget window
        base = WIDGET_AUTO_BASE(title ='RX Parameters')
        sb = WIDGET_BASE(base, /column, /frame)
        sb1 = WIDGET_BASE(sb, /row)
        sb2 = WIDGET_BASE(sb, /row)
        mw = WIDGET_MENU(sb1, prompt = 'Use Image Wide Covariance? ', $
                 list=['No','Yes'], /excl, default_ptr=0, $
                 rows=0, uvalue='iw', /auto)
        mw1 = WIDGET_MENU(sb2, prompt = 'Use As Matched Filter? ', $
                 list=['No','Yes'], /excl, default_ptr=0, $
                 rows=0, uvalue='use_match', /auto)
        sb = WIDGET_BASE(base, /column, /frame)
        sb1 = WIDGET_BASE(sb, /row)
        wp = WIDGET_PARAM(sb1, prompt='Kernel size (NxN) -- N = ', $
                 dt=2, xs=0, uvalue='N', default = 15, /auto)
        sb2 = WIDGET_BASE(sb, /row)
        wp1 = WIDGET_PARAM(sb2, prompt='Spatial extent of target (SxS) -- S = ', $
```

```
            dt=2, xs=0, uvalue='S', default = 3, /auto)
        sb = WIDGET_BASE(base, /column, /frame)
        ofw = WIDGET_OUTfm(sb, func='envi_out_check', uvalue = 'outf', /auto)

        result = auto_wid_mng(base)
        IF (result.accept EQ 0) THEN RETURN
        iw = result.iw
        N = result.N
        S = result.S

        IF result.use_match THEN BEGIN
          ENVI_SELECT, title='RX Input ENVI Spectral Library', $
                       file_type=4, fid=fidb, dims=dimsb, pos=posb

              IF (fidb EQ -1) THEN RETURN
        ENDIF

        ;N must be an odd integer
        IF (N MOD 2) NE 1 THEN BEGIN
          mstr = 'N must be an odd integer.'
          ENVI_ERROR, mstr, /warning
          RETURN
        ENDIF

        ;S must be odd integer
        IF (S MOD 2) EQ 0 THEN BEGIN
          mstr = 'S must be odd.'
          ENVI_ERROR, mstr, /warning
          RETURN
        ENDIF

        ;S must be less than N
        IF (S gt N) THEN BEGIN
          mstr = 'S must less than N.'
          ENVI_ERROR, mstr, /warning
          RETURN
        ENDIF


        IF result.use_match THEN BEGIN
          b = TRANSPOSE(ENVI_GET_DATA(fid=fidb, dims=dimsb, pos=pos))
          rx_doit, fid=fid, dims=dims, pos=pos, N=fix(N), $
              S=fix(S), b=b, IMAGE_WIDE=iw, out_name=result.outf.name, $
                       in_memory=result.outf.in_memory
          ENDIF ELSE BEGIN
          rx_doit, fid=fid, dims=dims, pos=pos, N=fix(N), $
                       S=fix(S), IMAGE_WIDE=iw, out_name=result.outf.name, $
                       in_memory=result.outf.in_memory
          endelse
        ENDIF
end
```

Appendix B

TreePlanter Code

```
PRO TreePlanter, xscale=xscale, yscale=yscale, $
        xorigin=xorigin, yorigin=yorigin, $
        xlocation=xlocation, ylocation=ylocation, $
        zlocation=zlocation, outputfilename=outputfilename, $
        pathname=pathname, scalestring=scalestring, height=height

IF not KEYWORD_SET(xscale) THEN xscale = .2
IF not KEYWORD_SET(yscale) THEN yscale = .2
IF not KEYWORD_SET(xorigin) THEN xorigin = 608
IF not KEYWORD_SET(yorigin) THEN yorigin = 769
IF not KEYWORD_SET(xlocation) THEN xlocation = 108.2
IF not KEYWORD_SET(ylocation) THEN ylocation = 79.1
IF not KEYWORD_SET(zlocation) THEN zlocation = 159.0
IF not KEYWORD_SET(outputfilename) THEN $
        outputfilename = "DIRSIG"
IF not KEYWORD_SET(pathname) THEN pathname = $
        "/dirs/home/dirsig/microscene/gdb/"

; scalestring = x and y scale to adjust tree size properly $
        - Format example:  ".6, .6"

;image must be a grayscale or indexed color image
tree_locations_image = reverse(READ_IMAGE(DIALOG_PICKFILE $
        (FILTER = ['*.tif'])),2)

sz = SIZE(tree_locations_image, /dim)

;tree_numbers = INTARR(sz[0],sz[1])

trees = WHERE(tree_locations_image NE 255)

openw,lun, STRCOMPRESS(STRTRIM(string(outputfilename+".odb") $
        ,2)), /get_lun,width=1000

PRINTF, lun, "OBJECT {"
PRINTF, lun, "        " + STRCOMPRESS(STRTRIM(string( $
        "GDB_FILENAME = ", pathname),2))
PRINTF, lun, "        UNITS = METERS"
PRINTF, lun, "        INSTANCES {"

tree_coords = fltarr(2)

FOR i=0, N_ELEMENTS(trees)-1 DO BEGIN
        xval = trees[i] mod sz[0]
        yval = sz[1] - 1 - trees[i] / sz[0]
        tree_coords[0] = xlocation + (xval - xorigin)*xscale
```

```
        tree_coords[1] = ylocation + (yorigin - yval)*yscale

        str = STRCOMPRESS(STRTRIM(string("#", i+1, " : x = ", $
                xval, " y = ", yval),2))
        PRINTF, lun, "              " + str

        str = STRCOMPRESS(STRTRIM(string("INFO = ", tree_coords[0], $
                ", ", tree_coords[1], ", ", zlocation, ", ", scalestring, $
                ", ", height+(randomu(seed)-.5)*.1, ", 0, 0, ", $
                fix(randomu(seed)*360)),2))
        PRINTF, lun, "              " + str

ENDFOR

PRINTF, lun, "       }"
PRINTF, lun, "}"
FREE_LUN, lun

END
```