Using Lidar to Geometrically-constrain Signature Spaces for Physics-based Target Detection

by

Michael S. Foster

B.S. United States Air Force Academy, 1999

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

23 August 2007

Signature of the Author _____

Accepted by _

Coordinator, Ph.D. Degree Program

Date

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

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

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

Dr. John R. Schott, Dissertation Advisor

Dr. David W. Messinger

Dr. Emmett J. Ientilucci

Dr. Carl Salvaggio

Dr. Joseph DeLorenzo

Date

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

Title of Dissertation:

Using Lidar to Geometrically-constrain Signature Spaces for Physics-based Target Detection

I, Michael S. Foster, 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

DISCLAIMER

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

Using Lidar to Geometrically-constrain Signature Spaces for Physics-based Target Detection

by

Michael S. Foster

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

Abstract

A fundamental task when performing target detection on spectral imagery is ensuring that a target signature is in the same metric domain as the measured spectral data set. Remotely sensed data are typically collected in digital counts and calibrated to radiance. That is, calibrated data have units of spectral radiance, while target signatures in the visible regime are commonly characterized in units of reflectance. A necessary precursor to running a target detection algorithm is converting the measured scene data and target signature to the same domain.

Atmospheric inversion or compensation is a well-known method for transforming measured scene radiance values into the reflectance domain. While this method may be mathematically trivial, it is computationally attractive and is most effective when illumination conditions are constant across a scene. However, when illumination conditions are not constant for a given scene, significant error may be introduced when applying the same linear inversion globally.

In contrast to the inversion methodology, physics-based forward modeling approaches aim to predict the possible ways that a target might appear in a scene using atmospheric and radiometric models. To fully encompass possible target variability due to changing illumination levels, a target vector space is created. In addition to accounting for varying illumination, physics-based model approaches have a distinct advantage in that they can also incorporate target variability due to a variety of other sources, to include adjacency effects, target orientation, and mixed pixels. Increasing the variability of the target vector space may be beneficial in a global sense in that it may allow for the detection of difficult targets, such as shadowed or partially concealed targets. However, it should also be noted that expansion of the target space may introduce unnecessary confusion for a given pixel. Furthermore, traditional physics-based approaches make certain assumptions which may be prudent only when passive, spectral data for a scene are available. Common examples include the assumption of a flat ground plane and pure target pixels. Many of these assumptions may be attributed to the lack of three-dimensional (3D) spatial information for the scene. In the event that 3D spatial information were available, certain assumptions could be levied, allowing accurate geometric information to be fed to the physics-based model on a pixelby-pixel basis. Doing so may effectively constrain the physics-based model, resulting in a pixel-specific target space with optimized variability and minimized confusion.

This body of work explores using spatial information from a topographic Light Detection and Ranging (Lidar) system as a means to enhance the fidelity of physics-based models for spectral target detection. The incorporation of subpixel spatial information, relative to a hyperspectral image (HSI) pixel, provides valuable insight about plausible geometric configurations of a target, background, and illumination sources within a scene. Methods for estimating local geometry on a per-pixel basis are introduced; this spatial information is then fed into a physics-based model to refine the forward prediction of a target in radiance space. The target detection performance based on this spatially-enhanced, spectral target space is assessed relative to current state-of-the-art spectral algorithms.

Acknowledgements

While sitting through Dr. John Schott's hyperspectral algorithms course, I noticed a common theme: almost all of the target detection algorithms were based solely on spectral features, with shape information being completely ignored. This is partially due to the fact that many of these algorithms were born out of the signal processing community, which typically does not deal with a spatial context. Another reason for this is because current remote imaging systems often lack the spatial resolution to produce many fully-resolved pixels on objects that are commonly considered targets, such as vehicles. Hence, the majority of the available shape information is lost when a target is imaged.

An emerging modality within the remote sensing community is topographic laser radar (or Lidar), which is capable of measuring the 3D spatial profile of a scene. Having worked with these systems previously, I was intrigued by the notion of fusing Lidar-based shape information with traditional hyperspectral imagery. While the process of searching for an object based on its "color" and shape is immediately intuitive to us as humans, teaching a computer to perform the same task is not trivial. To that end, I sought an intelligent means to combine Lidar and hyperspectral data for the purpose of improving automatic target detection performance.

During my time at RIT, I have been fortunate to have worked with many gifted friends and colleagues. These gifts ranged from being subject matter experts with the ability to teach, to possessing abundant patience while I rambled on about my latest idea or problem. I feel somewhat remiss in putting only my name on the title page of this dissertation, as my committee played a vital role during the research process. After teaching me the basics of remote sensing, Dr. Schott allowed me to run with an idea and had the foresight to keep me reigned in and moving in the right direction. Dr. David Messinger was a constant sounding board, providing clarity when many of my ideas lacked focus. Dr. Emmett Ientilucci retaught me everything I had forgotten from Dr. Schott's courses, as well as the finer details of physics-based target detection. Dr. Carl Salvaggio often served as the voice of reason and provided valuable technical insights. Dr. Joseph DeLorenzo took time out of his schedule to serve as the outside reader and provided an alternative viewpoint from a radar perspective.

In addition to my committee members, there are many other individuals whom I would like to thank. Scott Brown went out of his way to assist me in my DIRSIG simulations and taught me vector geometry fundamentals that would ultimately prove to be pivotal in my research. I would like to thank Steve Lach for always lending an ear and making valuable suggestions. Rolando Raqueño generously assisted me in running my time-intensive codes on the CONDOR cluster. Last but not least, thank you to Cindy Schultz who kept me out of trouble and was a constant source of encouragement.

Dedication

This dissertation is dedicated to my wife, without whom the work would not have been possible. Your faithful love and support have made all the difference in my life.

Contents

A	bstra	ıct		iv
Li	st of	Figur	es	xiv
\mathbf{Li}	st of	' Table	S	xiv
1	Intr	roduct	ion	1
2	Bac	kgrou	nd	10
	2.1	Hyper	rspectral Sensors	10
	2.2	Topog	graphic Lidar Sensors	12
	2.3	Hyper	rspectral Target Detection	16
		2.3.1	Atmospheric Compensation (Reflectance Domain) $\ldots \ldots \ldots$	16
		2.3.2	Forward Models (Radiance Domain)	18
			2.3.2.1 Healey's Method	18
			2.3.2.2 Extensions to Healey's Method	22
	2.4	Lidar	Target Detection	28
	2.5	Fused	Target Detection	31
	2.6	Curre	nt Algorithm Limitations	33
3	Var	iable S	Sensitivity Study	34
4	Me	thodol	ogy	41
	4.1	Lidar	Point Cloud Processing	43
		4.1.1	Sparse Shadow Feeler	43
		4.1.2	Ground Plane and Point Normal Estimation	49

			4.1.2.1 Terrain Extraction
			4.1.2.2 Point Normal Estimation
		4.1.3	Sparse Skydome Feeler
		4.1.4	Preliminary Target Point Extraction
			4.1.4.1 Z-windowing Above Ground Plane $\ldots \ldots 58$
			4.1.4.2 Spin-image Fundamentals
			4.1.4.3 Spin-image Target Detection
		4.1.5	Sparse Z -buffer
	4.2	Map (Creation
		4.2.1	3D Projection onto HSI FPA 79
		4.2.2	Shadow Map - K
		4.2.3	Illumination Angle Map - θ
		4.2.4	Skydome Map - F
		4.2.5	Pixel Purity Map - M
	4.3	Atmos	sphere Generation using MODTRAN
	4.4	Backg	round Radiance Cube Generation
	4.5	Const	rained Signature Space Generation
	4.6	Detect	tion Statistic
F	Dog	ulta or	ad Applyzia 102
9	nes		IU Allalysis 102
	5.1	Synth	etic Data $\ldots \ldots 102$
		5.1.1	Ideal Lidar, Ideal Hyperspectral
			5.1.1.1 Point Cloud Processing Results

			5.1.1.2	Map Creation Results	110
			5.1.1.3	Signature Space Generation Results	113
			5.1.1.4	Detection Results	118
		5.1.2	Ideal Lie	dar, Noisy Hyperspectral	131
		5.1.3	Noisy Li	idar, Ideal Hyperspectral	135
			5.1.3.1	Point Cloud Processing Results	136
			5.1.3.2	Map Creation Results	140
			5.1.3.3	Detection Results	143
		5.1.4	Noisy Li	idar, Noisy Hyperspectral	144
			5.1.4.1	Detection Results	144
		5.1.5	Ideal Lie	dar, Wide Angle Ideal Hyperspectral	145
			5.1.5.1	Map Creation Results	145
			5.1.5.2	Detection Results	147
	5.2	Real-w	vorld Dat	a	147
		5.2.1	Point C	loud Processing Results	148
		5.2.2	Map Cre	eation Results	151
	5.3	Conclu	usions .		154
6	Fut	uro W	ork		156
U	I U U	A 11			150
	6.1	Additi	ional anal	LYSIS	150
	6.2	Comp	utational	Efficiency	158
7	App	pendix			161
	7.1	Downy	welled Pa	th Radiance Estimation	161

7.2	Detect	tion Results Using Different Atmospheres	165
7.3	Impac	t of Range Quantization on Spin-Images	170
	7.3.1	Quantized Point Clouds	170
	7.3.2	Impact on Point Normals	173
	7.3.3	Impact on Spin-images	177

List of Figures

Target detection processes	3
Visible regime radiance path contributions	4
Illustration of radiometric model's geometric terms.	7
Illustration of mixed pixel term M	8
AVIRIS hyperspectral image cube.	11
False colored topographic map of RIT	13
Illustration of pushbroom Lidar system	14
(a) Spectral radiance triplet generated using MODTRAN (b) Reflectance	
curves for the target and background materials	35
Radiance curves modulated by shadowing term K . (a) Absolute (b) Peak	
normalized. \ldots	36
Radiance curves modulated by incident illumination angle term θ . (a) Abso-	
lute (b) Peak normalized.	37
Radiance curves modulated by shape factor term F for non-shadowed target.	
(a) Absolute (b) Peak normalized	38
Radiance curves modulated by shape factor term F for shadowed target. (a)	
Absolute (b) Peak normalized	39
Radiance curves modulated by pixel purity term M . (a) Open target (b)	
Concealed target.	39
Block diagram of signature space generation process. Target space variability	
introduced by modeling ${\cal N}$ atmospheres and modulating spatial feature maps	
according to uncertainty levels	42
	Target detection processes.Visible regime radiance path contributions.Illustration of radiometric model's geometric terms.Illustration of mixed pixel term M .AVIRIS hyperspectral image cube.False colored topographic map of RIT.Illustration of pushbroom Lidar system.(a) Spectral radiance triplet generated using MODTRAN (b) Reflectancecurves for the target and background materials.Radiance curves modulated by shadowing term K . (a) Absolute (b) Peaknormalized.Radiance curves modulated by incident illumination angle term θ . (a) Absolute (b) Peak normalized.(a) Absolute (b) Peak normalized.Radiance curves modulated by shape factor term F for non-shadowed target.(a) Absolute (b) Peak normalized.(a) Absolute (b) Peak normalized.(a) Absolute (b) Peak normalized.(b) Peak normalized.(curves modulated by shape factor term F for shadowed target.(a) Absolute (b) Peak normalized.(b) Peak normalized.(curves modulated by shape factor term F for shadowed target.(a) Absolute (b) Peak normalized.(b) Peak normalized.(curves modulated by pixel purity term M .(a) Open target (b)Concealed target.(b) Peak normalized.(curves modulated by pixel purity term M .(a) Open target (b)Concealed target.(curves modulated by pixel purity term M .(a) Open target (b)Concealed target.(curves modulated by modeling N atmospheres and modulating spatial feature mapsaccording to uncertainty levels.<

4.2	Illustration of sparse shadow feeler	44
4.3	Illustration of missing data in point cloud collected from single viewing geom-	
	etry. (a) Oblique view of cloud (b) nadir view of cloud	45
4.4	Illustration of using cylinders with spherical caps as an intersection volume.	46
4.5	Illustration of terms used in azimuthal boundary intersection test	48
4.6	Illustration of terms used in z-boundary intersection test	49
4.7	Illustration of candidate points (black) relative to point under test (red). (a)	
	azimuthal check (b) z check	50
4.8	Illustration of binary test using inverted conic structuring element [Sithole	
	2005]	51
4.9	Illustration of ground plane extraction process [Lach 2007]. (a) Original range	
	image (b) binary terrain extraction and (c) terrain range image	53
4.10	Illustration of sparse skydome feeler sample directions (red) and associated	
	quadrants. (a) Oblique view and (b) nadir view	54
4.11	Illustration of simple scene (a) and associated point cloud (b) for sparse sky-	
	dome feeler validation.	56
4.12	Illustration of z -windowing. (a) Threshold on original data and (b) threshold	
	after normalizing the ground plane	58
4.13	Illustration of spin-image parameters [Johnson 1997]	60
4.14	Illustration of gridding error and bilinear interpolation.	63
4.15	Example spin images from rubber duckie [Johnson 1997]	64
4.16	Block diagram of spin image target detection process	65
4.17	Major axis along M1 tank	68

4.18	3D cube and its surface normal vectors	69
4.19	Illustration of geometric consistency concept [Johnson 1997]	72
4.20	Three notional correspondence points between a model (a) and scene (b)	76
4.21	Illustration of vector geometry for planar projection.	82
4.22	Converting ${\bf q}$ from spatial to indexed coordinates	83
4.23	Illustration of shadow projected onto subgrid.	85
4.24	Integrated downwelled radiance distribution across the visible and near in-	
	frared spectrum	89
5.1	Image of Microscene I area.	103
5.2	Visualization of ideal 3D point cloud false colored by height. (a) Nadir view	
	and (b) oblique view	104
5.3	False color composite of ideal hyperspectral cube. (a) Actual HSI cube spatial	
	resolution - 1 m GSD and (b) high resolution reference - 0.125 m GSD	105
5.4	Target point extraction results: z-windowed ideal point cloud. \ldots	108
5.5	Target point extraction results: scene correspondences (yellow) and extracted	
	target points (red) for ideal point cloud.	109
5.6	Feature maps for ideal Lidar data. (a) Shadow map (b) incident illumination	
	map (c) shape factor map and (d) pixel purity map	111
5.7	Background map creation. (a) Sensor cube masked by pixel purity map and	
	(b) background radiance cube false color composite	114
5.8	Block diagram of signature space generation process for microscene	116

5.	9	Example signature spaces. (a) Space for pure pixel on target 3 and (b) space	
		for pure pixel on target 7	117
5.	10	Constrained (red) and unconstrained (blue) spectral signature spaces	117
5.	.11	Illustration of the best match spectra from several different signature spaces.	
		(a) Target 3 (b) target 7 and (c) target 6.	119
5.	12	(a) Inverse RMSE score map for ideal data set and (b) Truth map for resolved	
		(green) and subpixel (red) targets	120
5.	13	Ground truth maps. (a) Oversampled first material map and (b) Pixel purity	
		truth map	122
5.	14	False color composite of background end members. (a) Sand (b) tree (c) red	
		paint (d) grass and (e) shadow	123
5.	15	PB-OSP score map for ideal data set and (b) Truth map for resolved (green)	
		and subpixel (red) targets	123
5.	16	PB-OSP score maps for Ientilucci's global target space method. (a) Spatially-	
		cued detection scores, (b) non-cued detection scores, and (c) Truth map for	
		resolved (green) and subpixel (red) targets	126
5.	17	Block diagram for generating spectrally correlated noise	132
5.	18	Spectrally correlated noise: correlation matrices. (a) Input dark data and (b)	
		output noise vectors	133
5.	19	False color composite of hyperspectral cubes. (a) Original ideal cube and (b)	
		ideal cube with spectrally correlated noise added. \ldots	133
5.	.20	Detection score maps for ideal Lidar and noisy HSI cube. (a) Inverse RMSE,	
		(b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets	.134

LIST OF FIGURES

5.21	Target point extraction results: z-windowed noisy point cloud	137
5.22	Target point extraction results: scene correspondences (yellow) and extracted	
	target points (red) for noisy point cloud	138
5.23	Target point extraction results: scene correspondences (yellow) and extracted	
	target points (red) for ideal point cloud with wide z-window. \ldots . \ldots .	139
5.24	Feature maps for noisy Lidar data. (a) Shadow map, (b) incident illumination	
	map, (c) shape factor map, and (d) pixel purity map	142
5.25	Detection score maps for noisy Lidar and ideal HSI cube. (a) Inverse RMSE,	
	(b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.	143
5.26	Detection score maps for noisy Lidar and noisy HSI cube. (a) Inverse RMSE	
	(b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.	144
5.27	Feature maps for ideal data sets and wide angle HSI sensor. (a) Shadow map	
	(b) incident illumination map (c) shape factor map and (d) pixel purity map.	146
5.28	Detection score maps for ideal data sets and wide angle HSI sensor. (a) Inverse	
	RMSE, (b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red)	
	targets	147
5.29	(a) Range image of Leica point cloud (b) Image of pickup truck within Mi-	
	croscene I	148
5.30	Target point extraction results: z -windowed Leica point cloud	149
5.31	Target point extraction results: scene correspondences (yellow) and extracted	
	target points (red) for Leica point cloud.	150
5.32	Feature maps for Leica point cloud and notional HSI sensor. (a) Shadow map	
	(b) incident illumination map (c) shape factor map and (d) pixel purity map.	152

5.33	Range map for Leica point cloud.	153
7.1	DIRSIG tape5 card deck.	162
7.2	Tape5 card deck for two-run method	162
7.3	Tape5 card deck for single-run method	163
7.4	Downwelled path radiance estimation comparison	164
7.5	Path radiance curves for different atmospheres. (a) 12 km and (b) 5 $km.$	166
7.6	CAD model for SUV	171
7.7	Quantized point clouds for the SUV CAD model. (a) Nadir at 0.5 ns (b) slant	
	at 0.5 ns (c) nadir at 1.0 ns (d) slant at 1.0 ns (e) nadir at 2 ns and (f) slant	
	at 2 <i>ns</i>	173
7.8	Quantized normals for the SUV CAD model based on $n=3$. (a) Nadir at 0.5	
	ns (b) slant at 0.5 ns (c) nadir at 1.0 ns (d) slant at 1.0 ns (e) nadir at 2 ns	
	and (f) slant at 2 ns	175
7.9	Quantized normals for the SUV CAD model based on $n=6$. (a) Slant at 0.5	
	ns (b) slant at 1.0 ns and (c) slant at 2 ns	176
7.10	Basis points, normals, and spin-images. (a) Perfect model cloud (b) perfect	
	spin-image (c) nadir at 2.0 ns , $n=6$ (d) nadir at 2.0 ns , $n=6$ spin-image (e)	
	Slant at 2 ns, $n=6$ (f) Slant at 2 ns, $n=6$ spin-image (g) Slant at 2 ns, $n=3$	
	and (h) Slant at 2 ns , $n=3$ spin-image	179

List of Tables

2.1	Range of atmospheric and geometric parameters.	20
5.1	Pixel-level PB-OSP performance comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data	128
5.2	Pixel-level PB-OSP false alarm comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data	128
5.3	Object-level PB-OSP false alarm comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data (green = correct,	
	red = incorrect)	129
5.4	Point cloud processing run times (approximate)	130
7.1	Pixel-level PB-OSP performance comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data using 12 km visi-	
	bility atmosphere	168
7.2	Pixel-level PB-OSP false alarm comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data using 12 km visi-	
	bility atmosphere	168
7.3	Object-level PB-OSP false alarm comparison between fused approach and	
	Ientilucci's method (HSI only), based on ideal Lidar/HSI data using 12 km	
	visibility atmosphere (green = correct, red = incorrect)	168
7.4	Pixel-level PB-OSP performance comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data using 5 km visi-	
	bility atmosphere	169

7.5	Pixel-level PB-OSP false alarm comparison between fused approach and Ien-	
	tilucci's method (HSI only), based on ideal Lidar/HSI data using 5 km visi-	
	bility atmosphere	169
7.6	Object-level PB-OSP false alarm comparison between fused approach and	
	Ientilucci's method (HSI only), based on ideal Lidar/HSI data using 5 km	
	visibility atmosphere (green = correct, red = incorrect)	170

Chapter 1

Introduction

An ever-present challenge to the remote sensing community is the prediction of target signatures under varying illumination conditions and levels of contamination and/or concealment. Illumination conditions may vary temporally and spatially within a given scene, particularly in the presence of significant cloud cover or highly varying terrain. Furthermore, sub-pixel targets may manifest themselves in sensor radiance space as a linear or nonlinear combination of target and background radiance signatures. Current methods for performing spectral target detection fall into two fundamental categories, atmospheric compensation and physics-based model methodologies. Compensation relies on using in-scene ground truth information and/or radiative transfer codes such as the MODerate resolution TRANsmittance code (MODTRAN) to estimate atmospheric parameters [Klempner et al. 2006. These parameters may then be used to invert the effect of the atmosphere across the image, therein converting the image from the radiance domain into the reflectance/emissivity domain. Upon transforming the collected sensor data into reflectance space, various target detection algorithms may be used to compare the estimated scene reflectance information to one or more known target reflectance spectra and ultimately arrive at a detection probability map. Conversely, physics-based modeling approaches aim to project a target's known reflectance signature into radiance space through physics-based forward models. Instead of producing a single projection of the target into the radiance domain, the invariant approach produces multiple radiance space signatures of the target, as seen through a variety of probable illumination conditions. This set of radiance space signatures is referred to as a target vector space and in theory, characterizes all of the possible ways that a target might appear in an image. As such, running a detection algorithm on the target space should perform equally well regardless of illumination state. For this reason, physics-based modeling approaches are often referred to as *invariant* [Healey and Slater 1999]. A scene's background vector space may be determined empirically from the scene itself or through *a priori* knowledge of background constituents. Invariant approach target detection algorithms compare scene-based radiance information to target and background vector spaces to produce a detection probability map. The accuracy of the invariant approach hinges upon model accuracy, sensor calibration fidelity, and sufficient separability between the target and background radiance vector spaces. Figure 1.1 shows a top-level block diagram of the two visible regime target detection processes described above. The left half of the diagram represents the atmospheric inversion process with detection occurring in the reflectance domain, while the right is illustrative of the invariant approach.

A significant advantage that the invariant approach has over atmospheric inversion methods is its ability to incorporate in-scene variability due to a variety of sources. A radiometric model which may be used to generate target vector spaces can be written as [Schott 1997]

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[\left(E'_s(\lambda)\tau_1(\lambda)\cos\theta + FE_d(\lambda) \right) \tau_2(\lambda) \frac{r(\lambda)}{\pi} + L_u(\lambda) \right] d\lambda \quad \left[\frac{W}{m^2 \operatorname{sr} \mu m} \right] .$$
(1.1)

This equation predicts the spectral radiance $L_p(\lambda)$ at a sensor based on a target reflectance for a specific geometry and atmosphere. Direct solar $E_s(\lambda)$, downwelled $E_d(\lambda)$, and upwelled radiance $L_u(\lambda)$ path contributions in Eq. (1.1) are illustrated in Figure 1.2 along paths A, B, and C respectively. Emissive radiance terms and target contamination due to in-scene



Figure 1.1: Target detection processes.

adjacency contributions have been purposefully omitted for simplification; as such, this model is appropriate only for predicting pure target signatures in the visible, near infrared, and shortwave infrared regimes. $\beta_p(\lambda)$ is the spectral response of the sensor for spectral band p. $E'_s(\lambda)$ is the direct exoatmospheric solar irradiance. $\tau_1(\lambda)$ and $\tau_2(\lambda)$ are the sun-target and target-sensor path transmissions, respectively. θ is the angle at which direct solar irradiance is incident upon the target and may therefore be calculated as the angle between the target surface normal and solar illumination vector. The solar illumination vector is defined as the unit vector pointing from the sun to the target. It is common to approximate θ as the solar zenith angle σ' (*i.e.*, declination angle) relative to a scene's global coordinate system normal when a target's normal orientation is unknown. F is a scalar term used to modulate the amount of downwelled (*i.e.*, scattered) solar irradiance $E_d(\lambda)$ upon a target. F provides a



Figure 1.2: Visible regime radiance path contributions.

means of capturing the fact that objects immediately adjacent to a target may partially block the full skydome to which the target is exposed. The reflectance spectrum of the target is characterized by $r(\lambda)$. Finally, $L_u(\lambda)$ is the upwelling solar radiance that has been scattered back into the sensor field of view without interacting with the target or background.

One disadvantage of using Eq. (1.1) as a physical model is that it fails to incorporate variability due to mixed pixels. It is important to note that despite the fact that the model itself does not account for mixed pixel variability, target detection methods based on this model have demonstrated varying degrees of effectiveness in detecting subpixel targets [Liu and Healey 2006; Ientilucci 2005]. These detection algorithms will be discussed in further detail in Sections 2.3.2.1-2.3.2.2. A mixed pixel occurs when a sensor's pixel, as projected to the ground, contains more than one material constituent (*i.e.*, target and background). Pixel homogeneity is often inversely related to a sensor's pixel element(s) as projected to the ground, also known as ground instantaneous field of view (GIFOV). In order to achieve high spectral resolution, it is common for hyperspectral sensors to have large pixels and corresponding GIFOVs. The Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), an airborne hyperspectral sensor, has a GIFOV on the order of 20 meters [Jet Propulsion Laboratory 2006]. As such, most AVIRIS scenes over land contain many mixed pixels. Recent work by Messinger proposes that target spectrum variability due to mixed pixels may be incorporated into spectral radiance models by treating the sensor-reaching radiance as a fractional mixture of the target and background [Messinger 2005]. A slight variation of Messinger's mixing model is proposed as

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[K E'_s(\lambda) \tau_1(\lambda) \cos \theta + F E_d(\lambda) \right] \tau_2(\lambda) \left[M \frac{r_t(\lambda)}{\pi} + (1 - M) \frac{r_b(\lambda)}{\pi} \right] + L_u(\lambda) \, \mathrm{d}\lambda$$
(1.2)

Notice that two reflectance spectra $r_t(\lambda)$ and $r_b(\lambda)$ are present in Eq. (1.2) enabling the target and background respectively to contribute to the overall radiance signature. With the inclusion of the fractional mixing term M and shadowing term K, flexibility in the forward modeling of the target vector space is increased. As long as the target radiance space is sufficiently separable from the background, successful detection of sub-pixel targets may be realized. Obviously, some lower limit must be imposed on the amount of background content that is included in the linear mixing construction. If not, the case of complete target obscuration would result in overlap in the target and background. A target detection algorithm based on this physical model is described in Section 4.6.

All of the variables within Eq. (1.2) with the exception of $r_t(\lambda)$ may be varied during the target space generation process associated with physics-based forward modeling. Doing so effectively accounts for all of the ways that a given target might appear in the radiance domain. For example, $\cos \theta$ may be varied from 0 to 1 to account for all possible orientations of the target surface normal relative to the sun's position in the sky. While varying these terms globally may be beneficial when trying to build a thorough target space, doing so may also make the span of the target vector space unnecessarily large for a given pixel.

The objective of this research is to explore means of optimizing target vector spaces on a per-pixel basis based on 3D geometric scene information, as provided by a georegistered, cotemporal topographic Lidar sensor. Using only a Lidar's spatial information (*i.e.*, return intensity is ignored), it may be possible to infer local spatial information for a hyperspectral scene and therefore constrain the range over which certain terms from Eq. (1.2) are varied for a given pixel when generating a pixel-specific target vector space. An implicit assumption in this approach is that the Lidar system's spatial sampling frequency is much higher than that of the hyperspectral sensor.

A useful way of characterizing the terms in Eq. (1.2) is to group them into two fundamental categories: the variables are either inherently spectral or geometric in nature. Obviously, a Lidar system's spatial information may be used to constrain only those geometric terms, to include K, θ , F, and M. Clearly, the locations of shadows within a scene is driven by the spatial extent of objects relative to the sun's position in the sky. Figure 1.3 illustrates the geometric nature associated with θ and F. Figure 1.3a illustrates the solar zenith angle σ' , as measured from a global coordinate system normal, and the incident illumination angle θ



Figure 1.3: Illustration of radiometric model's geometric terms.

relative to the local surface normal. Figure 1.3b is a simple illustration of the shape factor term F, used to attenuate the downwelled irradiance incident upon the target. Obviously if a target is surrounded by tall, opaque structures, it will be exposed to less than a full hemispherical skydome. It is important, however, to realize that F may not be equal to the fractional portion of the sky dome visible to the target. This is due to the fact that downwelled radiance is often not uniformly distributed across the sky dome.

The final geometric term M is illustrated in Figure 1.4. The diagram shows a focal plane array's (FPA) pixel boundaries projected onto a nadir scene consisting of a tank in a grass field. For this scenario, the tank is considered the target, while the grass is background. The projected pixels, as indicated by the dashed lines, are filled with varying fractional amounts of target, resulting in pixel-specific M values.

As stated earlier, an aim of this research is to use Lidar information to estimate the geometric terms in Eq. (1.2). In what may seem as an initial contradiction, estimation of M



Figure 1.4: Illustration of mixed pixel term M.

requires a priori knowledge of target location relative to a hyperspectral FPA's projected pixels, implying the task of target detection has already been performed. However, a relevant question that arises is, "What is the confidence level in the target detection process based on solely the Lidar information?" The answer to this question is largely driven by the ratio of the Lidar's spatial sampling resolution relative to the spatial features of the target in question. One could make the argument that current Lidar sensors, operating at sufficient standoff ranges to be considered *remote*, do not have a small enough post spacing to perform high confidence level target detection. However, relative to a hyperspectral sensor, a Lidar sensor may have sufficient spatial sampling to provide an estimate of mixing fraction levels, therein providing useful information to a physics-based model.

3D spatial information from a Lidar point cloud may be used to estimate values for the geometric terms in Eq. (1.2) on a pixel-by-pixel basis. It is important to recognize that the estimates for these geometric terms will have associated uncertainties. This work accounts for these uncertainties by incorporating variability about the geometric estimates when generating a spectral target space for each HSI pixel. The inclusion of variability about the geometric terms may still allow for the resulting spaces to be constrained relative to invariant methodologies based on solely passive hyperspectral information. Therein lies the advantage in the constrained approach: the addition of 3D spatial information allows for pixel-specific target space optimization and may ultimately result in better target detection performance, particularly so for challenging target scenarios.

Chapter 2

Background

The bulk of this chapter focuses on target detection methods associated with opticalregime remote sensing modalities and combinations thereof. Individual modalities to be discussed briefly in Sections 2.1 and 2.2 include topographic passive hyperspectral and topographic Lidar sensors. For a more rigorous treatment of these types of sensors, see [Schott 1997] and [Measures 1984]. Sections 2.3 and 2.4 provide a sampling of published target detection techniques associated with hyperspectral and Lidar sensing modalities, respectively. While there is extensive literature available on single-modality target detection algorithms, little work has been published on combining the information from multiple types of data sources into a single target detection product. The term *fused* is a common term within the remote sensing community and is used in a myriad of contexts. For the purposes of this paper, *fused* will refer to multi-modal data products. Section 2.5 describes previous work relating to fused target detection. Finally, Section 2.6 provides a brief synopsis of this body of background literature and identifies potential areas for advancement in target detection.

2.1 Hyperspectral Sensors

Traditional electro-optical (EO) imaging systems integrate across a single, large spectral bandpass to create a grayscale image of a scene. Multi- and hyperspectral systems are imaging spectrometers that spectrally sample a scene to generate multiple grayscale images corresponding to each of the spectrometer's spectral bandpasses. As such, spectral data sets are described as *cubes* due to the inclusion of an additional spectral dimension. Figure 2.1 shows an AVIRIS data cube of Moffett Field, California [Jet Propulsion Laboratory 2006]. The top surface of the cube is a grayscale image from the sensor's first spectral channel, centered at 400 nm. The depth of the cube corresponds to 224 spectral bands, extending out to 2.5 μm , with each channel's resolution being approximately 10 nm. Also, notice that the spectral dimension exhibits gaps due to atmospheric absorption lines. The number of spectral



Figure 2.1: AVIRIS hyperspectral image cube.

bandpasses dictates whether a spectral sensor is considered to be multi- or hyperspectral. Typically, multispectral sensors are defined as having roughly ten or fewer broad spectral channels, while hyperspectral sensors may have anywhere from ten to hundreds of contiguous, narrow channels [Schott 1997].

2.2 Topographic Lidar Sensors

Topographic Lidar systems are rapidly becoming more common within the remote sensing community. The primary use for this type of system is the generation of high resolution terrain information. Prior to the introduction of topographic Lidar systems, Digital Elevation Maps (DEM), generated by ground-based surveys, were a common representation of 3D geospatial terrain data. While the cell spacing of a US Geological Survey (USGS) DEM may be on the order of 10 m with vertical accuracies of equal to or better than 5 m, current Lidar systems are capable of measuring scenes with an order-of-magnitude greater accuracy [U.S. Geological Survey 2006]. Furthermore, the amount of manpower, time, and associated expense in generating a topographic product may be greatly reduced using Lidar. The image in Figure 2.2 shows a topographic mapping of the RIT campus. The false coloring of the image corresponds to scene displacement relative to the global coordinate system's z-axis. Thus, regions of constant color represent level areas, while color gradients correspond to sloped regions in the scene. Much like passive, visible regime images, Lidar scenes are also subject to illumination shadowing, as indicated by the black pixels. Smoothed regions about the edges of buildings and trees branches result from interpolation between points from the original 3D point cloud.

Lidar sensors use one or more pulsed lasers to illuminate a field of view. By syncing the sensor's illumination source(s) and on-board clock to a common trigger, the round-trip flight time t_r of a reflected pulse may be measured. Figure 2.3 shows a notional pushbroom Lidar system flying over an urban scene. The colored terrain indicates the cumulative Field of View (FOV) that has been mapped by the sensor. Assuming that the return photons



Figure 2.2: False colored topographic map of RIT.

only undergo a single bounce – i.e. photons travel directly to a feature in the scene, undergo reflection, and propagate directly back to the sensor – the time-of-flight measurement and speed of light c may be used to derive the range d to the scene feature according to

$$d = \frac{c \cdot t_r}{2} \quad [\mathrm{m}] \quad . \tag{2.1}$$

In addition to range measurements, some Lidar sensors also measure the intensity of the return pulse as a means to capture a scene's reflectivity at the source wavelength. For the purposes of this research, the intensity information provided by a Lidar sensor will be



Figure 2.3: Illustration of pushbroom Lidar system.

ignored.

One disadvantage commonly associated with Lidar systems is the fact that they tend to be photon-starved at the receiver aperture. While the truthfulness of the generalization is largely dependent upon a Lidar system's design and operating characteristics, Eq. (2.2) illustrates the fundamental relationships that govern the power at a Lidar's receive aperture P_S [Farmer 2001].

$$P_S = \frac{P_0 \tau_T F_b \rho_t \cos \theta A_R \tau_R}{\pi R_{mx}^2} (S \tau_{atm})^2 \quad [W]$$
(2.2)

 P_0 is the initial, peak output power from the laser source, while τ_T is the transmission of the transmit optics. F_b is the beam fill factor, describing the relative fill ratio of the incident laser beam spot to the detector's GIFOV. If the beam spot is smaller than the GIFOV, then $F_b = 1$. Otherwise, assuming a square detector element, $F_b = GIFOV^2/A_s$, where A_s is incident beam area. ρ_t is the integrated reflectance of the GIFOV at the laser wavelength.
For fully resolved targets, ρ_t is the reflectance of the target material at the source laser's operating wavelength. θ is the incident illumination angle of the source beam relative to the GIFOV's integrated surface normal. A_R and τ_R are the area and transmission of the receive optics, respectively. R_{mx} is the maximum allowable distance from which return photons are received, as determined by range gating settings. S and τ_{atm} are the Strehl and atmospheric transmittances respectively, the distinction between the two being that Strehl transmittance captures the effect of atmospheric turbulence while atmospheric transmittance τ_{atm} is purely a function of atmospheric constituents.

Much like imaging spectrometers, Lidar sensors come in a variety of collection formats, to include line scanners, pushbroom scanners, and framing arrays. Regardless of the collection methodology, these sensors are ideal for making high resolution, 3D topographic maps of scenes. The transverse angular resolution r_t of a Lidar system, sometimes referred to as *post spacing*, is dictated by the same mechanisms that limit the resolution of passive imaging systems. However, a Lidar system's resolution in the range dimension r_r is driven by the source's laser pulse width pw, as indicated by

$$r_r = \frac{pw \cdot c}{2} \quad [m] \quad . \tag{2.3}$$

The resolutions of airborne Lidar systems may vary dramatically depending upon the intended application. One example of an existing system is the Optech ALTM 3100, which has transverse and range resolutions of 1 m and 0.25 m respectively when the system is flown at an altitude of 2 km [Optech Incorporated 2004]. Furthermore, multiple passes of the same scene may allow for higher post spacings in the resulting co-registered point clouds.

As such, a primary assumption in this work, that a Lidar system might be able to feed spatial information to a hyperspectral sensor on a per-pixel basis, may not be unreasonable. As an extreme example, consider the scenario of the Optech ALTM 3100 flown over a scene at the same time as an AVIRIS overflight. Given the 20 m size of AVIRIS's GIFOV, it is not an ideal hyperspectral sensor for target detection applications; however, the subpixel, spatial information provided by the Lidar might allow for reasonable hyperspectral target detection performance.

2.3 Hyperspectral Target Detection

The field of spectral target detection is both expansive and diverse. Summarizing the progression of target detection from its conception as a by-product of signal detection theory to the current state-of-the-art exceeds the scope of what is immediately relevant to this body of work. The following taxonomy was developed by Manolakis and Shaw to provide a framework for characterizing hyperspectral algorithms: *(i)* method for describing target/background variability, *(ii)* purity of data pixel being tested, and *(iii)* detection statistic model used to test a given pixel [Manolakis and Shaw 2002]. The following sections describe hyperspectral algorithms that are relevant to this body of research using this taxonomy.

2.3.1 Atmospheric Compensation (Reflectance Domain)

Inherent to the hyperspectral target detection process is the requirement to convert measured data and a known target signature to the same metric domain. Traditionally, digital count or radiance domain information collected at a sensor platform is compensated, therein

transforming the data to the reflectance domain. In doing so, scene-derived reflectance information may be compared to measured or known target spectra of interest. If ground truth information is available, global compensation may be applied to the imagery via the Empirical Line Method (ELM) [Schott 2007]. Ground truth, usually in the form of two or more calibration panels, provides reflectance-domain reference points from which a linear relationship with the observed radiance may be derived. Generally, the reflectance domain references approximately span the range of reflectance values in order to maximize linear fit accuracy. If ground truth is not available, one may assume reflectance values for scene pixels and invert the image, often at the expense of accuracy. The primary disadvantage of compensating imagery via ELM-based methods is that the correction is applied globally to an image, based on known or assumed ground truth. Therefore, if illumination conditions vary spatially, then error is introduced through the inversion process. Alternately, compensation algorithms, such as Fast Line-of-sight Atmospheric Analysis of Spectral Hypercubes (FLAASH), Atmosphere Correction Now (ACORN), and Atmospheric Removal program (ATREM), use look-up tables generated by a radiative transfer model, such as MODTRAN, to estimate optimal inversion Perkins *et al.* 2005; Miller 2002; Lanorte *et al.* 2002]. These algorithms iteratively vary atmospheric parameters in the radiative transfer model until a quality match between the measured and predicted scene radiance for a given pixel is achieved [Klempner et al. 2006]. The algorithms perform reasonably well across a variety of scenes; however, their accuracy may degrade in the presence of dropout and/or saturated pixels and between-band calibration errors. Furthermore, many of the compensation algorithms make assumptions about local scene geometry which may or may not be correct, to include the assumption of a flat local ground plane and no adjacency effects. As an alternative to atmospheric compensation, target detection based on physical models is introduced in Section 2.3.2.

2.3.2 Forward Models (Radiance Domain)

Section 2.3.1 describes an overall approach for converting radiance domain information, as collected by a sensor, to the reflectance domain for the purpose of comparing test pixels to a known target reflectance spectrum. Forward models, or physics-based approaches, aim to invert this problem by predicting how a target pixel might appear in the radiance domain. Obviously, a considerable amount of variability must be incorporated into a radiance-domain description of a target. Therein lies the function of, and burden placed on, the physics-based radiometric model. Additionally, proper characterization of the target and/or background spaces generated by the model has a considerable impact on the fidelity of the approach. These issues are discussed in detail in the following sections.

2.3.2.1 Healey's Method

The invariant method was first introduced by Healey and Slater in 1999 [Healey and Slater 1999]. Target and background spaces are radiance-domain representations of target and background reflectance spectra, respectively. Ideally, these spaces encompass all of the different ways that a given reflectance spectrum might manifest itself in the radiance domain. As such, spectral variability must be incorporated into these spaces.

A space may be constructed using a known target reflectance spectrum $R(x, y, \lambda)$ and a

physical model, such as

$$L(x, y, \lambda) = T_u(z_g, z_\nu, \theta_\nu, \lambda) R(x, y, \lambda) \\ \times \left[KT_d(z_g, \theta_0, \phi_0, \lambda) E_0(\lambda) \cos(\theta_0) \\ + \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi/2} E_s(\theta, \phi, \lambda) \cos(\theta) \sin(\theta) d\theta d\phi \right] \\ + P(z_g, z_\nu, \theta_\nu, \phi_\nu, \lambda)$$
(2.4)

Similar to the previous physical models introduced in Chapter 1, the terms T_u and T_d are the upward and downward atmospheric transmittances, respectively. z_g specifies a matte surface's ground elevation, as defined by a polar coordinate system consisting of elevation and azimuthal angles θ_{ν} and ϕ_{ν} , respectively. z_{ν} designates a sensor platform elevation, based on the same coordinate system. E_0 , E_s , and P are direct, downwelled, and upwelled solar radiances. Finally, K is a binary term used to model target shadowing due to external occlusion of the target by opaque bodies along the direct path between the target and sun.

Several of the geometric terms, z_g , θ , and K, in Eq. (2.4) are assumed to be unknown. Therefore, the predicted range of variability associated with these terms is incorporated into the corresponding space. The spectral variables associated with atmospheric constituents and scattered solar radiance may also be largely unknown. The effect of the atmosphere on the final radiance signatures reaching a remote sensing platform is significant and varies spatially across a scene. The MODTRAN atmospheric model is used to account for the radiance-space variability imparted by the atmosphere. Hence, MODTRAN outputs spectrally-dependent vectors for E_0 , E_s , P, T_u , and T_d . The range associated with these terms is largely dependent upon collection location, in-scene geometries, and time of year. A single combination of the terms may be used to effectively model one possible atmosphere for a scene.

Once the appropriate range for the geometric and spectral terms have been established, these values are fed into Eq. (2.4) to create a radiance-domain space. Each variation in these variables results in an additional radiance vector being added to the space. Obviously, varying the geometric and spectral terms over a wide range of values results in a vast number of radiance vectors. Table 2.1 is an example of a relatively simple set of input parameters used in generating a radiance-domain space for a single reflectance spectrum. Notice that

Scene Parameters	Values
$H_20 \text{ (cm)}$	0.88, 1.44, 3.11, 4.33
$O_3 (\text{atm cm})$	0.07, 0.11, 0.14
$O_2 (\text{atm m})$	8407.9, 8604.0, 9453.2, 10536.8
$CH_4 (\text{atm cm})$	0.85, 0.86, 0.87
$N_2O \text{ (atm cm)}$	0.799, 0.202, 0.209, 0.214
CO (atm cm)	0.064, 0.065, 0.067, 0.070
$CO_2 (\text{atm m})$	15.23, 16.17, 17.33, 17.76
Aerosol type	rural, urban, maritime, desert
Solar-zenith angle (°)	5, 15, 25, 35, 45
Sensor altitude (km)	1, 2, 3, 4, 5, 6, 7

Table 2.1: Range of atmospheric and geometric parameters.

the upper eight rows of the table directly impact the spectral output of MODTRAN, while the lower two rows are geometric in nature. The gases in the table are prevalent atmospheric constituents that create absorption bands. The solar-zenith angle and sensor altitude allow for varying scene geometries. This combination of input variables ultimately results in 17,920 physically valid estimates of how a single target reflectance spectrum might appear in radiance-space [Healey and Slater 1999].

Eq. (2.4) may be used to generate either a target or background space. Using this

physical model to predict a background space may be problematic in that characterization of the background requires accurate reflectance spectra for every background material in the scene. Thus, this approach may be impractical for large, complex scenes. An alternative approach to background space generation is presented in Section 2.3.2.2.

Once target or background material's space has been established, the dimensionality W of the space is reduced by selecting a set of N orthonormal basis vectors $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_N$ to form a subspace that adequately spans the respective space. Any normalized scene radiance vector $\hat{\mathbf{L}} = \mathbf{L}/\|\mathbf{L}\|$ may be modeled as a linear combination of basis vectors for a given material subspace according to

$$\hat{\mathbf{L}} = \sum_{j=1}^{N} \alpha_j \mathbf{m}_j + \mathbf{n} \quad .$$
(2.5)

The subspace basis vectors associated with material i are denoted as R_i . Then, the Gaussian maximum likelihood that a scene radiance vector, as projected onto R_i , belongs to the material associated with R_i is calculated according to

$$P(\hat{\mathbf{L}}|R_i) = \frac{1}{(2\pi)^{0.5W} |\Sigma|^{0.5}} \exp(-0.5\mathbf{D}^t \Sigma^{-1} \mathbf{D}) \quad , \tag{2.6}$$

where Σ is the covariance matrix of a zero-mean Gaussian random vector **n** and

$$\mathbf{D} = \hat{\mathbf{L}} - \sum_{j=1}^{N} \alpha_j \mathbf{m}_j \quad .$$
 (2.7)

D is a vector which quantifies the difference between the normalized test pixel $\hat{\mathbf{L}}$ and its estimation, computed as a linear combination of R_i . The maximum likelihood estimate for

 $\alpha_1, \alpha_2, \ldots, \alpha_N$ may be solved for by taking the derivative of $\mathbf{D}^t \Sigma^{-1} \mathbf{D}$ with respect to each α_i , setting the N expressions to zero, and solving for $\alpha_1, \alpha_2, \ldots, \alpha_N$.

In addition to a target detection method, the test in Eq. (2.6) may also be applied as a material identification algorithm. However, a significant shortcoming in Healey's original implementation is that Eq. (2.5) only accounts for fully resolved pixels. This issue, as well as others, is addressed in further extensions to Healey's method described in the following section.

2.3.2.2 Extensions to Healey's Method

Since Healey and Slater's introduction of the invariant approach, several revisions to the method have been published. In 2001, Pan published a paper on extending the invariant approach from 2D matte to 3D targets under unknown conditions. Pan constructs a target space using the Digital Image and Remote Sensing Image Generation (DIRSIG) simulation tool and spectrally facetized, 3D models of a target in a variety of poses [Pan and Healey 2001; Brown 2006]. Another significant modification to Healey's approach made by Pan was the incorporation of a mixed pixel model. Unresolved targets are modeled via the following hypothesis test:

$$H_0: \mathbf{y} = \gamma B + \mathbf{n}$$

$$H_1: \mathbf{y} = \alpha T + \gamma B + \mathbf{n}$$
(2.8)

The null hypothesis H_0 predicts that a pixel **y** may be best modeled as a linear combination of background basis vectors B, plus a residual noise term **n** that is assumed to be meancentered, Gaussian white noise. The alternate hypothesis H_1 models a pixel as a linear combination of both target basis vectors T and background, plus the noise term. Each column of B and T is a basis vector from the respective subspace group [Pan and Healey 2001]. Instead of using a single likelihood metric, as in Eq. (2.6), a Gaussian maximum likelihood ratio is computed as

$$\Lambda(\mathbf{y}) = \frac{p_1(\mathbf{y}|H_1)}{p_0(\mathbf{y}|H_0)} \quad .$$
(2.9)

This ratio may then be compared to standard F-statistics to determine the outcome of the binary hypothesis test for a given pixel.

As a final contribution to the invariant method, Pan introduces the concept of using in-scene features to construct a background subspace. Assuming the scene is composed of predominantly background pixels with a few sub-pixel targets, Singular Value Decomposition (SVD) of the image is used to estimate background basis vectors. An l-band radiance image consisting of p pixels may be represented as

$$Y = UDV^t \quad . \tag{2.10}$$

Y is an $l \times p$ matrix where each column is a radiance vector from the hyperspectral image. U and V are the eigenvectors of YY^t and Y^tY , respectively, while D is a diagonal matrix of singular values sorted in descending order. Finally, the background may be represented by the first r columns of U [Pan and Healey 2001]. While the accuracy in the background representation is generally directly related to r, the increase in accuracy gained by adding additional basis vectors generally falls off beyond a certain point. Furthermore, using r < lbasis vectors may result in better characterization of the background if sensor noise artifacts are characterized by the low singular values.

The obvious problem with using spectral pixels from a scene to predict a background

space is that all of the pixels may not be solely background. Inclusion of target radiance vectors in the background space is referred to as *background contamination*. As such, care must be taken when deriving background basis vectors from in-scene radiance vectors. If a pixel containing target contributes to the background subspace, then the performance of subsequent detection statistics, based on the contaminated background basis vectors, is likely to suffer. This is because most detection algorithms seek to suppress pixels that appear more like the background than target. Hence, if the background description is contaminated with one or more target pixels, then those same target pixels are likely be suppressed by the detection algorithm.

Thai further extends the in-scene background subspace generation technique by proposing a method for background contamination mitigation [Thai and Healey 2002]. Prior to applying SVD to the scene, target candidate pixels are removed from the data set. The process of removing target candidates consists of two steps. First, a target space is generated using the physics-based model and one or more known target spectra under a variety of geometric/atmospheric conditions. Next, each scene pixel \mathbf{y} is projected onto the target subspace T via

$$\gamma = \frac{\|T^t \mathbf{y}\|^2}{\mathbf{y}^t \mathbf{y}} \quad . \tag{2.11}$$

The term γ serves as a measure of similarity, with values close to 1 indicating that the projection of a pixel onto the target subspace is large. Therefore, pixels with γ values that are close to 1 are removed from the image prior to running SVD to compute the background subspace.

Ientilucci presents an alternative in-scene technique for preventing background subspace

contamination. In order to avoid such contamination, one may augment the scene's radiance space data vectors with the target space basis vectors derived from physical modeling. One caveat in this approach is the augmented target basis vectors must be in the native space (e.g. radiance) and have the same number of dimensions as the original image. If a scene consists of 100 pixels and a target subspace consists of four basis vectors, then all 104 vectors would be supplied to a background basis vector-finding technique. In doing so, the output basis vectors are likely to contain the four, supplied target basis vectors in addition to the desired background basis vectors. Since the augmented target basis vectors are known, they may be removed from the final set of background basis vectors. This approach effectively shields these scene-based techniques from target contamination [Ientilucci 2006].

Intilucci introduces another variation on the forward modeling approach through the use of a different physical model,

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[\left(E_{s.new}(\lambda) \tau_1(\lambda) + F E_d(\lambda) \right) \tau_2(\lambda) \frac{r(\lambda)}{\pi} + L_u(\lambda) \right] d\lambda \quad \left[\frac{W}{m^2 \operatorname{sr} \mu m} \right].$$
(2.12)

This model is essentially the same as Eq. (1.1), allowing for the suppression of the downwelled solar radiance by multiplying the $E_d(\lambda)$ term by a scalar F, ranging from 0 to 1. In contrast, Healey's model, *i.e.* Eq.(2.4), integrates the downwelled radiance over the entire hemisphere above a target, therein not accounting for partial occlusion of the sky dome. It should be noted, however, that Eq. (2.12) does not have a binary term for extinguishing the direct solar loading in the event of target shadowing; rather, Ientilucci's approach suppresses the direct solar term via the $\cos \sigma_{new}$ term described below.

In addition to implementing a different forward model, Ientilucci also treats varying

target orientation differently than Healey. Instead of using Healey's method of varying the sun's elevation above a target, Ientilucci effectively fixes the sun's position and varies the illumination percentage on the target via

$$E_{s_new}(\lambda) = \frac{E_s(\lambda)}{\cos \sigma'} \cos \sigma_{new} \quad , \tag{2.13}$$

where $E_s(\lambda) = E'_s(\lambda) \cos \sigma'$, $\sigma_{new} = \sigma' - \sigma_{rot}$, and σ_{rot} is a user-supplied angle of rotation relative to the solar declination angle σ' . While the difference in this illumination construction may initially seem trivial, the benefit of using this method is that the solar loading terms, both direct and scattered, may be more accurate due to sun-to-target paths relative to MODTRAN's modeling of atmospheric layers [Ientilucci 2005].

Finally, instead of testing scene pixels using a Gaussian maximum likelihood ratio test, Ientilucci introduces the Physics Based-Structured InFeasibility Target-detector (PB-SIFT), which is described in greater detail in Section 4.6.

One assumption that is common to the invariant methodologies described thus far is the assumption of well-behaved noise in the pixel detection models. In Healey's basic method and the subsequent sub-pixel detection schemes, pixel estimate residuals are presumed to be zero-mean, Gaussian noise. Liu discards this assumption and uses nonparametric methods to characterize the irregular distributions associated with these residual vectors [Liu and Healey 2006]. To establish nonparametric estimates of the distributions, two discriminant features are exploited: (1) the magnitude of the residual vector for a given pixel and (2) the direction of the residual vector. Probability density functions (PDFs) for target, background, and mixed residual 2D spaces are estimated based on a training data set. Next, these PDFs are

used in combination with in-scene background basis vectors to generate optimal projection operators for discriminating between pure/mixed target spaces and the background. Compared to orthogonal subspace projection operators, the non-parametric projection operators are claimed to increase separability between the target and background, thus resulting in improved detection performance. To this end, Liu introduces a Nonparametric Likelihood Ratio Test which was demonstrated to nominally outperform invariant methods based on both traditional generalized likelihood ratio and residual generalized likelihood ratio tests for a variety of real and synthetic data sets.

While all of the invariant methods mentioned thus far in this section use a sub-pixel detection model for hypothesis testing, none allow for mixtures between target and background constituents in their forward predictions of radiance domain signatures. Liu proposes that the union of target and background spaces may be useful in a sub-pixel detection scheme; however, spectral *mixing* of the two spaces is not described [Lui and Healey 2003]. The forward prediction of subpixel targets requires knowledge of target and one or more background spectra that comprise a pixel. Also, the nature of the mixing between a pixel's constituents may not be linear, as is indicated by Eq. (1.2) [Messinger 2005]. For example, a mixed pixel consisting of a target material obscured by a camouflage net could possibly be a nonlinear mixture between the two materials. Such might be the case if incident photons were to transmissively pass through the net, reflect off the target, and transmissively pass through the net again along their return path to a sensor. Messinger proposes that adaptation of physical models might be necessary for accurate characterization of different subpixel scenarios, such as concealed and/or contaminated targets.

Recent work by Chandra demonstrates that a technique based on the invariant method

may be used to estimate the orientation of local surfaces. A space created by the direct solar irradiance $E'_s(\lambda)$, downwelled irradiance $E_d(\lambda)$, and upwelled radiance $L_u(\lambda)$ component contributions in Eq. (1.1) may be treated as a coupled space, due to the common dependence of the terms upon sensor geometry and atmospheric conditions [Chandra and Healey 2006]. SVD is applied to each of the radiance component spaces separately to generate three subspaces. As a result, any measured radiance vector may be represented as a linear combination of the component basis vectors associated with $L_s(\lambda)$, $L_d(\lambda)$, and $L_u(\lambda)$. A constrained nonlinear optimization method based on sequential quadratic programming techniques is used to solve for the linear combination that minimizes the error between a pixel vector and its associated estimate. Finally, this solution may then be used to estimate a surface's normal to within 10°, based on a test set of 15 fully resolved target panels. While the method does show promise, it was proven to fail for targets in shadow and cannot be used to estimate the mean surface orientation for mixed pixels [Chandra and Healey 2006].

2.4 Lidar Target Detection

Before going into specific target detection algorithms, it may be useful to distinguish between automatic target detection (ATD) and automatic target recognition (ATR). Target detection entails a coarse, high-level search of a Lidar point cloud to locate 3D regions that are likely to contain target candidates. Target recognition then takes location cues from the detection process and refines the assessment by either placing an identifier label from a known target library on a location or discarding the location as a non-target. The Lidar-based processes in this body of work are focused solely on the task of target detection; however, many of the Lidar detection techniques described in Section 4.1.4.3 could be extended to incorporate recognition [Vasile and Marino 2004]. This work does address target recognition through secondary spectral processing which benefits from fused, preliminary spatial information provided by a topographic Lidar sensor.

Most Lidar detection algorithms were born out of computer and robotic vision applications. Lidar data sets provide a variety of sensor-native features to which detection schemes may be applied. Algorithms may be applied to 1D range dimension information, a 2D intensity image, and/or a raw 3D point cloud. Furthermore, interpolation and false coloring of a 3D point cloud may be used to generate yet an additional presentation of local shape information in the form of a false-colored range image. Such a presentation of Lidar data is useful for performing ground plane extraction and may be considered a form of feature extraction. False-colored range images and ground plane retrieval are discussed further in Section 4.1.2.

The vast majority of Lidar target detection may be classified as template matching. Template matching consists of extracting a 1-, 2-, or 3-D silhouette from a data set and comparing it to a known target template. Retrieved silhouette shapes are often characterized by one or more features such as invariant moments, orthogonal polynomials, and Fourier metrics [Zhou and Sapounas 1997]. These scene-based features are then compared to features produced from target models. Accurate silhouette extraction is often difficult in real imagery due to a variety of reasons, such as lack of resolution, unknown articulation of the target, and target occlusion [Wellfare and Norris-Zachery 1997]. However, genetic algorithms and neural networks have been trained to detect such shape features in 3D data, demonstrating robust performance for smaller scenes [Sadjadi 1994; Snorrason *et al.* 1995,]. Surface fitting approaches seek to recognize primitive features in 3D data sets. Surface primitives are simple geometric features, such as planes or cylinders, which, when taken in a collective geometric context, may be matched to a target template. A main advantage with surface fitting is that complex features can often be characterized by a small number of surface primitives, therein reducing computational expense. Until recently, accurate surface primitive matching was considered to be limited to short-range systems. This was primarily due to increased susceptibility to range noise associated with long-range Lidar systems [Zheng et al. 2001,]. However, a surface fitting approach called planar patch segmentation has proven to be robust in the presence of noise [Cobzas and Zhang 2001]. Additionally, 3D feature-grouping methodologies have been shown to be robust in the presence of moderate occlusion [Stein and Medioni 1992].

Thus far, a few challenges associated with Lidar-based target detection have been highlighted. First, an algorithm must be computationally efficient when addressing the issue of lack of *a priori* knowledge of target articulation and pose. Adding degrees of freedom to a target model can quickly make the task of recognition intractable from a combinatoric standpoint. Second, target detection algorithms for remote sensing applications must scale well for large scenes. Finally, an ideal detection algorithm must be robust in the presence of noise, occlusion, and clutter [Carmichael and Huber 1999].

Instead of template matching, individuals at Massachusetts Institute of Technology Lincoln Laboratory demonstrated the use of spin-image techniques to perform both target detection and recognition in Lidar point clouds [Vasile and Marino 2004]. A spin-image is a 2D parameter space histogram that captures the majority of local shape information present in a 3D scene. Several aspects of the spin-image approach make it an attractive option for performing target detection/recognition. These aspects include robust detection performance for large scenes with one or more targets in arbitrary orientations, as well as graceful degradation in the presence of occlusion and clutter. According to Vasile, spin-image template matching is among the most promising detection techniques for processing 3D point clouds [Vasile and Marino 2004]. The spin-image shape matching process was developed by Johnson for robotic vision applications and is explained in detail in Sections 4.1.4.2-4.1.4.3 [Johnson 1997].

A common theme present in these Lidar target detection publications is the need for high resolution data sets for robust target detection performance. Since many long-range Lidar systems cannot provide high resolution data, *e.g.* 200 points on a target, without making a large number of passes over a scene of interest, alternate methods for target detection may need to be explored. Multi-modal fusion provides an interesting option for salvaging such lower resolution shape information and is discussed in the following section.

2.5 Fused Target Detection

As stated at the beginning of the chapter, *fused* has many meanings within the remote sensing community. In this document, the term is defined as a joint information product from multi-modal data sources. Furthermore, a taxonomy describing the level at which information fusion occurs needs to be established. Fusion can occur at the following fundamental levels: (*i*) data-level, (*ii*) pixel-level, and (*iii*) decision-level [Ratches *et al.* 1997,]. Data-level fusion refers to the incorporation of sensor-native information from multiple sensors combined into a single product, such as the texture associated with a passive image being applied to a DEM. Pixel-level fusion occurs when images are overlaid relative to their respective pixels to create an enhanced image product. The fusion of hyperspectral imagery with high resolution panchromatic data sets has been proven to be an effective way to sharpen hyperpsectral products [Gross and Schott 1996]. Finally, decision-level implies that independent detection outputs from two or more sensor modalities are combined. An example of this would be using boolean operators to combine the binary target detection maps from a hyperspectral and SAR sensors. One advantage of decision-level fusion is that it allows for easy integration of mode-specific algorithms across many different phenomenologies. The Multi-Sensor Image Cueing (MUSIC) tool is a decision-level detection suite capable of ingesting data from synthetic aperture radar, infrared, and electro-optical images [Rodvold and Patterson 2002]. While decision-level fusion is likely to be the easiest form of fusion to implement, whether or not combining information at the decision-level is an optimum use of platform data is arguable. Rather, data-level fusion may provide the most information about a scene for use in subsequent processing. An example of successful data-level fusion for remote sensing target detection is demonstrated in [Baum and Rak 1991]. A simultaneous CO_2 Lidar/passive LWIR sensor was flown by Massachusetts Institute of Technology's Lincoln Laboratory. The co-registered range information from the Lidar sensor was shown to be an effective means for mitigating false alarms from the LWIR channel. Additionally, the fusion of statistical signals from Lidar and forward-looking infrared sensors is described in [Hannon and Shapiro 1990; Kostakis et al. 1999,]. Unfortunately, little has been published with regards to data-level fusion of 3D spatial and spectral information.

2.6 Current Algorithm Limitations

In summary, forward modeling approaches to hyperspectral target detection have shown promise in detecting targets in a variety of atmospheric/geometric environments. While invariant methods have been proven to be effective in detecting 3D subpixel targets under certain conditions, the performance of the invariant approach degrades as the difficulty of the target scenario increases. For example, a shadowed, subpixel target on a steep slope might not be detected by the invariant approach due to inadequate characterization of the target space. Current physics-based forward models are limited due to lack of knowledge of 3D scene geometry. On the other hand, while Lidar target detection algorithms have proven to be widely effective for short-range applications with high sampling densities, in many cases the performance of the detection algorithms degrades rapidly with increasing stand-off range, post spacing, and range quantization levels. The novelty in this research is the method by which relatively low resolution Lidar information will be used to enhance the forward prediction of spectral signatures for hyperspectral target detection. As such, this work is an example of data-level fusion of 3D spatial information with invariant spectral algorithms in order to study the potential for improved target detection performance.

Chapter 3

Variable Sensitivity Study

A preliminary sensitivity study was accomplished in order to estimate the relative impact of several terms in a physics-based, forward radiometric model. Prior to addressing this study, a new physical model is introduced as

$$L_p(\lambda) = \left[K \frac{L_s(\lambda)}{\cos\sigma} \cos\theta + FL_d(\lambda) \right] \left[Mr_t(\lambda) + (1 - M)r_b(\lambda) \right] + L_u(\lambda) \quad . \tag{3.1}$$

This model is similar to the mixing model presented in Eq. (1.2). Primary differences are the exclusion of an integrated sensor response function and expressing the ground-reflected direct and scattered solar terms as path radiances $L_s(\lambda)$ and $L_d(\lambda)$. All path radiance terms may be derived from MODTRAN via a process described fully in Section 4.3. It is important to recognize that MODTRAN uses many input parameters, to include platform-to-scene range and pointing geometry, as well as atmosphere type, date, time of day, and geolocation information, to estimate the solar zenith angle σ and path transmissions $\tau_1(\lambda)$ $\tau_2(\lambda)$. As such, σ and the path transmissions are applied to the output, ground-leaving path radiance terms that appear in Eq. (3.1) according to

$$L_s(\lambda) = E'_s(\lambda)(\cos\sigma)\tau_1(\lambda)\tau_2(\lambda)/\pi$$

$$L_d(\lambda) = E_d(\lambda)\tau_2(\lambda)/\pi$$
(3.2)

Recall that in Eq. (2.4), Healey uses a binary K term preceding the direct solar loading term [Healey and Slater 1999]. Similar to Healey's construction, the K term is used to modulate the amount of direct sunlight incident upon a scene material. In contrast, this research treats K as a real number, ranging from 0 to 1. In doing so, partial shadowing of a target and/or background material within a pixel may be effectively modeled.

Much of the proposed research hinges upon the accurate prediction of target signatures through the use of forward modeling. A preliminary sensitivity study was conducted to estimate the relative impact of several of the terms in Eq. (3.1) upon the overall radiance vector, namely the terms which may be estimated from Lidar data, K, θ , F, and M. A path radiance vector triplet, plotted in Figure 3.1(a), was created using MODTRAN for a sensor flying at 12.5 km with a nadir viewing geometry in a midsummer latitude atmosphere with 23 km visibility.



Figure 3.1: (a) Spectral radiance triplet generated using MODTRAN (b) Reflectance curves for the target and background materials.

The target and background materials used in the experiments were a flat gray sheet metal and freshly mowed grass, respectively. The associated reflectance spectra for these materials were measured with a hand-held, spectral radiometer and are shown in Figure 3.1(b). The spectra indicate that the target material is a relatively low reflector across the visible and near infrared (VNIR) spectrum, while the background is considerably brighter past the red edge at 750 nm.

The first experiment involved observing how modulating the direct solar radiance via K effects the overall radiance vector \mathbf{x} , the results of which are presented in Figure 3.2. These spectra were generated by fixing all of the terms in Eq. (3.1) and varying K though values of 1.0, 0.75, 0.5, and 0.25. θ , F, and M were held to values of 0°, 0.8, and 1.0, respectively. While Figure 3.2(a) illustrates the expected changes in magnitude, the plots in Figure 3.2(b) have been magnitude normalized (*i.e.*, $\frac{\mathbf{x}}{\|\mathbf{x}\|}$) to observe changes in spectral shape. These curves show how the suppression of direct solar component causes the cumulative radiance to converge on the upwelled radiance distribution. This effect is amplified by the fact that the target material has low reflectance in the VNIR.



Figure 3.2: Radiance curves modulated by shadowing term K. (a) Absolute (b) Peak normalized.

The next experiment is based on varying the solar illumination angle θ and produces similar effects to the shadowing experiment; however, the magnitude falloff is not linear due the cosine term. Figure 3.3 illustrates the effect of varying θ from 20° to 80° in 20° increments.



Figure 3.3: Radiance curves modulated by incident illumination angle term θ . (a) Absolute (b) Peak normalized.

In varying the shape factor F, two different cases were run, resulting in the plots shown in Figures 3.4 and 3.5. These cases were intended to illustrate whether or not the impact of the downwelled contribution was correlated with the magnitude of the direct solar term. The first case varies F between 0.25 and 1.0 in increments of 0.25 with a non-shadowed target (*i.e.*, K = 1), while the second case is based on a target in soft shadow (*i.e.*, K = 0.3).

Per the downwelled radiance plot in Figure 3.1(a), one would expect the downwelled contribution to be greatest at small values of λ due to Rayleigh scattering. The plots indicate that, regardless of shadowing conditions, F has little impact on the resulting spectrum for



Figure 3.4: Radiance curves modulated by shape factor term F for non-shadowed target. (a) Absolute (b) Peak normalized.

targets with low reflectivity across the visible spectrum in a clear atmosphere.

The final experiment involved varying the linear mixing term M between 0.2 to 1.0 in increments of 0.4 to create different mixed spectra. This experiment also involved two separate cases to show how the output mixtures can vary dramatically based on illumination levels. The first case models a target that is fully illuminated and exposed to the majority of the skydome (*i.e.*, K = 1 and F = 0.8). Since the target and background reflectance spectra exhibit a high degree of spectral contrast, the resulting mixtures in the radiance domain follow suit, as shown in Figure 3.6(a). The opposite should also hold true.

The second case is meant to simulate a concealed subpixel target. Specifically, the target is unoccluded; however, it is in a soft shadow and is exposed to little of the skydome (*i.e.*, K = 0.3 and F = 0.3). 3.6(b) illustrates how suppression of the ground-leaving radiance contributions can significantly nullify the spectral contrast in the resulting radiance mixtures.



Figure 3.5: Radiance curves modulated by shape factor term F for shadowed target. (a) Absolute (b) Peak normalized.



Figure 3.6: Radiance curves modulated by pixel purity term M. (a) Open target (b) Concealed target.

In conclusion, these simple experiments are intended to provide the reader with an in-

tuitive sense of how local scene geometry can impact measured radiance signatures. While the relative importance of any one of the various parameters will be specific to local features within the scene, the experiment does imply that an accurate estimate of the shadowing term K is of utmost importance. The impact of an error in one's estimate of the incident illumination angle should be less severe for small θ values due to the cosine effect. Errors in the estimate of the shape factor F should have the least impact in atmospheres with high visibility (*i.e.*, $\geq 20 \text{ km}$), while error tolerance in M is inversely related to target/background spectral contrast and directly related to local illumination conditions. With these factors in mind, Chapter 4 explains how 3D Lidar data may be used to estimate local geometric information and subsequently be transformed into 2D feature maps for the purpose of optimizing the output of a radiometric mixing model.

Chapter 4

Methodology

This chapter describes techniques for creating pixel-specific radiance signature spaces, as described in Chapters 1 and 3. The overall process consists of the following fundamental steps:

- 1. Lidar point cloud processing
 - (a) Shadow feeler
 - (b) Extract ground points and calculate associated point normals
 - (c) Skydome feeler
 - (d) Spin-image target point detection
- 2. Create feature maps via projection onto HSI FPA
- 3. Generate MODTRAN atmospheres
- 4. Estimate background radiance cube
- 5. Propagate steps 2-4 into forward radiance model

First, the Lidar point cloud is processed to produce several different products. Three processes entail labeling every point in the cloud with a feature descriptor of interest, to include whether or not the point is in shadow, the fraction of the skydome above a point that is visible, and if the point is likely to be a target sample. The Lidar point cloud is also processed to extract point samples associated with the ground and their associated normal vectors. All of these processes are described in greater detail in Sections 4.1.1-4.1.4.

The second step involves projecting feature points of interest onto the hyperspectral FPA to produce a 2D image or map of the feature of interest. These maps constitute local estimates of the various spatial terms in the forward radiance model, Eq. (1.2). The third step involves using MODTRAN to generate one or more atmospheres that are believed to accurately represent the atmosphere present in the sensor radiance cube. Fourth, a background radiance cube is empirically derived using the sensor radiance cube and a Lidar target purity map. This process is described in greater detail in Section 4.4. Finally, the various maps, atmospheres, and background radiance cube are fed into Eq. (1.2) to produce a predicted signature space for every pixel in the sensor radiance cube. Steps 2-5 of this process are illustrated in Figure 4.1.



Figure 4.1: Block diagram of signature space generation process. Target space variability introduced by modeling N atmospheres and modulating spatial feature maps according to uncertainty levels.

4.1 Lidar Point Cloud Processing

The following section demonstrates several different ways that a Lidar point cloud may be processed. These methods are optimized for sparse data clouds and perform relatively well when data is only available from a single viewing geometry. Each method assigns a feature "label" to every point in the 3D cloud. It should be clear that the accuracy of these labels is directly related to point location accuracy, mean post spacing, and the number of viewing geometries from which the cloud was generated.

4.1.1 Sparse Shadow Feeler

The term *shadow feeler* refers to a ray tracing technique that is used to determine the portions of a geometric model that are visible, relative to a single point illumination source. In doing so, shadowed regions relative to the illumination source may be derived. Traditionally, shadow feeler techniques have been developed for facetized 3D models and are commonly found in 3D rendering software packages [Foley *et al.* 1992]. For this application, a *sparse shadow feeler* technique is introduced to estimate which points in a 3D cloud are likely to be in shadow. The sun's position in the sky for any given location, date, and time, is widely available in solar ephemeris tables. Hence, the spatial geometries inherently defined in a 3D point cloud and solar illumination direction provide a mechanism for predicting shadowed points.

Various problems may arise when attempting to implement a sparse shadow feeler on a sparse point cloud. This is primarily due to the fact that a single 3D point has no spatial extent; hence, ray tracing intersections with surround points are improbable. Instead of searching for a ray/point intersections, the sparse shadow feeler renders a 3D sphere about potential points of intersection and searches for ray/sphere intersections, as is illustrated in Figure 4.2. The radius of these spheres is defined as one half of the mean post spacing for



Figure 4.2: Illustration of sparse shadow feeler.

the Lidar point cloud. For the case of point samples from a single planar surface, the volume subtended by rendered spheres about the coplanar points forms a reasonable approximation to the actual solid surface geometry.

When dealing with point clouds collected from a single viewing geometry, it is common to have gaps in the data due to self-occlusion or external obscuration. Such gaps can be problematic when applying a sparse shadow feeler based solely on ray/sphere intersections, as described above. Figure 4.3 illustrates this problem. Notice that a ray extends from the point under test in the direction of the sun. Despite that fact the ray passes through the building, the ray does not intersect any of the rendered spheres associated with the other points in the cloud; hence a sparse shadow feeler based solely on ray/sphere intersections would incorrectly label the test point as not being in shadow. The proposed sparse shadow feeler circumvents this problem by assuming that all points are sampled from solid geometry that extends to the



Figure 4.3: Illustration of missing data in point cloud collected from single viewing geometry. (a) Oblique view of cloud (b) nadir view of cloud.

ground plane in the -z direction. In other words, a cylinder with a spherical cap is rendered about the intersection candidate points instead of a sphere. Another way to describe this volume is the union of a sphere centered on an intersection candidate point and a cylinder that extends from the intersection point to the ground plane in the -z direction. The radius of the sphere r_s is equal to half of the mean post spacing for the Lidar point cloud. The radius of the cylinder is also equivalent to that of the sphere, while the height of the cylinder is equivalent to the z-height of the intersection point relative to the ground plane. The use of cylinders with spherical caps as an intersection volume is illustrated in Figure 4.4.

The description of the sparse shadow feeler thus far has been purely conceptual. For the sake of computational speed, the actual implementation of the sparse shadow feeler algorithm does not render volumes about points; rather, it defines boundary conditions about candidate



Figure 4.4: Illustration of using cylinders with spherical caps as an intersection volume.

intersection points based on these volumes. The mathematical development of the algorithm defines the point under test for being in shadow as \mathbf{p} , having an (x, y, z) coordinate location of (x_p, y_p, z_p) . The solar zenith or declination angle σ is assumed to be constant across the scene, as is \vec{s} , which is defined as the unit vector originating at the global coordinate system origin pointing towards the sun. An intersection candidate point is defined as \mathbf{t} , with corresponding coordinates of (x_t, y_t, z_t) . The azimuthal component of \vec{s} is defined as \vec{s}_{ϕ} and represents the solar pointing direction projected onto the (x, y) plane. Similarly, azimuthal components of the points \mathbf{p} and \mathbf{t} are denoted as \mathbf{p}_{ϕ} and \mathbf{t}_{ϕ} , respectively.

$$\mathbf{p}_{\phi} = [p_x, p_y]$$

$$\mathbf{t}_{\phi} = [t_x, t_y] \quad . \tag{4.1}$$

$$\vec{s}_{\phi} = [\vec{s}_x, \vec{s}_y]$$

The following equations,

$$\alpha = (\mathbf{t}_{\phi} - \mathbf{p}_{\phi}) \cdot \frac{\vec{s}_{\phi}}{\|\vec{s}_{\phi}\|}$$
$$\vec{\alpha} = \alpha \frac{\vec{s}_{\phi}}{\|\vec{s}_{\phi}\|}$$
$$\vec{\beta} = \mathbf{t}_{\phi} - \mathbf{p}_{\phi} - \vec{\alpha}$$
$$\|\vec{\beta}\| \stackrel{?}{\leq} r_{s}$$
(4.2)

test whether the azimuthal component of the ray from **p** to the sun intersects the azimuthal boundary volume about candidate intersection point **t**. That is, if the magnitude of $\vec{\beta}$ is less than or equal to the radius of the sphere/cylinder boundary volume, then a volume intersection has occurred in the azimuthal dimension of the data. Note that an additional test, provided by Eq. (4.3), is required to test for a volume intersection in the z dimension of the data. The azimuthal boundary intersection test is illustrated below in Figure 4.5.

An intersection test perpendicular to the (x, y) plane is accomplished according to

$$d = \|\mathbf{p}_{\phi} - \mathbf{t}_{\phi}\|$$

$$h = d \tan(90 - \sigma) \qquad . \tag{4.3}$$

$$h \stackrel{?}{\leq} (t_z + r_s - p_z)$$

If the solar zenith angle σ is known, this second test may be implemented to query the height at which the line extending from **p** to the sun crosses the line from the ground to **t** in the +z direction, as is illustrated in Figure 4.6. This test is critical in accounting for situations such as the one illustrated in Figure 4.3 where data is missing from vertical scene structure.



Figure 4.5: Illustration of terms used in azimuthal boundary intersection test.

As such, it is referred to as a z-boundary intersection test. The combination of the two tests in Eqs. (4.2-4.3) is equivalent to testing for a ray intersection with a sphere-capped cylinder volume, as described above. If both equation conditions are met, then point \mathbf{p} is labeled as being in shadow.

Obviously, testing every point in the cloud as a candidate intersection point is computationally inefficient. Simple tests may be employed relative to a point under test \mathbf{p} to determine candidate intersection points. First, an azimuthal check is employed such that only points that are in the same (x, y) quadrant as the sun, relative to \mathbf{p} , are considered. Additionally, only points with higher z-coordinate values are considered. These checks are illustrated in Figure 4.7 below.

Again, the output of the sparse shadow feeler is a binary feature label on every point in



Figure 4.6: Illustration of terms used in z-boundary intersection test.

the 3D point cloud for the scene. While the set of 3D scene points and corresponding feature labels does provide shadow information for the scene, additional processing is required before such information may be incorporated into a spectral signature space. Specifically, the points must be projected into the FPA in order to provide integrated shadow information for each pixel. The projection process for creating a shadow map is described fully in Section 4.2.2.

4.1.2 Ground Plane and Point Normal Estimation

When processing raw Lidar data, knowledge of the ground plane can be invaluable. For the purposes of this research, ground plane estimates will be used in two different ways. First, it is used to estimate the orientation of incident illumination upon target within the scene, as described in Section 4.2.3. An implicit assumption in this estimate is that dominant target geometries are oriented parallel to the ground plane. Furthermore, in order to estimate the



Figure 4.7: Illustration of candidate points (black) relative to point under test (red). (a) azimuthal check (b) z check.

illumination angle, normal vectors associated with the ground plane are also estimated. The second use of the ground plane is to define the region of the point cloud where target points are likely be located. This is described in further detail in section Section 4.1.4.1.

4.1.2.1 Terrain Extraction

There are numerous published techniques for extracting the ground plane from a 3D Lidar point cloud. Unfortunately, common sensor artifacts such as range quantization and photon returns from multiple scattering events can make extraction of the true ground plane difficult. A full discussion of the strengths and weaknesses of the various terrain extraction methods is beyond the scope of this research; however, an in-depth review by Sithole is a suggested resource [Sithole 2005].

The extraction method used in this research is a morphological slope-based filter similar
to that described by Vosselman [Vosselman 2000]. This technique assigns a binary label to points within a cloud (*i.e.*, ground/non-ground) based upon local height differentials relative to 3D structuring element, a planimetric inverted funnel of radius r. For a point under test v_i , the funnel is raised in the +z direction until it touches the test point. If any points within the 3D span of the funnel have a z-value that is less than that of the test point, then the test point is labeled as non-ground. This concept is illustrated in Figure 4.8. The gray



Figure 4.8: Illustration of binary test using inverted conic structuring element [Sithole 2005].

conic regions in the figure represent the funnel test at different points within a cloud. Points that are labeled as non-ground are filled with gray, while terrain points have white centers. Some points at the center of tall, non-ground features may initially be labeled as ground by this process (notice white-centered points on roof top); therefore, a secondary heuristic may need to be applied to remove such spurious ground points. Implementation of Vosselman's extraction algorithm is based upon a local neighborhood function defined as

$$\phi_{N,r}(v_i) = \left\{ \begin{array}{ll} 0 \quad \forall \ v_j \in N \quad \exists \quad \Delta h(v_i, v_j) > \Delta h_{max}(d(v_i, v_j)) \\ 1 \quad else \end{array} \right\} \quad . \tag{4.4}$$

The neighborhood function tests N points v_j within the 3D funnel span. If any point v_j within the set N has a relative height difference $h(v_i, v_j)$ that is greater than the expected maximum height differential $h_{max}(d(v_i, v_j))$, then the test point v_i is labeled as non-ground. Typically, both r and $h_{max}(d(v_i, v_j))$ are empirically derived based on point cloud characteristics such as mean post spacing and general topography.

The most common output of a ground plane extraction method is a tag on the original cloud points that are believed to be samples from the terrain. That is, the ground points are not interpolated to a regular grid or altered from their original (x, y, z) coordinate values. Obviously, if non-ground features were present in the original point cloud, then the resulting cloud of ground points will have gaps where the non-ground features have been removed. Interpolation to a regular grid serves to fill in these gaps at the expense of adding uncertainty to the resulting ground point cloud. Interpolation to a grid is also useful for visualization of the 3D data as an image, as gray levels or a color table may be applied to represent the z dimension of the data, as is shown in Figure 4.9 [Lach 2007].

4.1.2.2 Point Normal Estimation

Besides knowledge of the 3D locations of ground plane points, an estimation of the normal vector associated with each point must be made. While the notion of a point normal may be initially non-intuitive, recall that ground plane points are either direct or interpolated



Figure 4.9: Illustration of ground plane extraction process [Lach 2007]. (a) Original range image (b) binary terrain extraction and (c) terrain range image.

spatial samples of the actual ground surface. A technique based on eigenvector decomposition is useful in estimating the normal vector associated with a 3D distribution of points. This process begins with selecting a point of interest \mathbf{p} , for which a normal vector will be estimated. Next, the Euclidean distance to all of the points in the cloud is calculated relative to \mathbf{p} . kclosest points are chosen, with the caveat that all of the k points cannot be co-linear within a user-specified tolerance. In practice, k should be a relatively small number so that only local variability is considered when estimating the normal. 3D coordinates of the k points are stored in an array C, consisting of 3 columns and k rows, where each row of C contains the 3D coordinates of a single point. Next, the sample covariance of C is computed, resulting in the 3×3 matrix Σ_C . Finally, the normalized eigenvector associated with the smallest eigenvalue of Σ_C is the preliminary estimate for the point normal \mathbf{n} . Given the two possible orientations of \mathbf{n} , a check of the pointing direction is necessary to ensure that the vector is pointing towards the sensor. This check consists of taking the scalar product of \mathbf{n} with the vector pointing from \mathbf{p} to the sensor location. If the scalar product is positive, then the normal is properly oriented. Otherwise, **n** should be multiplied by a scalar value of -1.

4.1.3 Sparse Skydome Feeler

A sparse skydome feeler algorithm is derived as a generalization of the sparse shadow feeler logic. Instead of probing in a single direction (*i.e.*, solar direction), the hemisphere above a point is probed many times in search of sphere/cylinder volume intersections (see Section 4.1.1). The proposed sparse skydome feeler samples the skydome in 30° azimuthal increments and 15° increments in zenith, for a total of 72 sample directions, as illustrated in Figure 4.10. Recall that the percentage of the sky that is visible for any given point in a



Figure 4.10: Illustration of sparse skydome feeler sample directions (red) and associated quadrants. (a) Oblique view and (b) nadir view.

scene is referred to as the shape factor. A point's shape factor is calculated as the weighted

average of its respective skydome feelers, according to

$$F = 1 - \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i} \quad , \tag{4.5}$$

where f_i is the binary skydome feeler output for the *i*-th sample direction. w_i is the weight assigned to *i*-th sample direction and is equal to the solid angle subtended by the *i*-th sample quadrant. The solid angle may be calculated via

$$\Omega = \int_{\phi_1}^{\phi_2} \int_{\theta_1}^{\theta_2} \sin\theta \,\mathrm{d}\theta \mathrm{d}\phi \quad , \tag{4.6}$$

where θ is defined as the declination angle as measured from the +z axis and ϕ is the angular measure of azimuth. This weighting factor is necessary because the impact of the skydome samples across the hemisphere varies with zenith angle. As is evident in Figure 4.10, the solid angle subtended by a quadrant is directly related to the quadrant's mean zenith angle. Therefore, samples taken at larger zenith angles should be given more weight in the overall calculation of F.

Sparse Skydome Feeler Validation

An experiment was conducted to validate the output of the sparse skydome feeler code. Specifically, the experiment was intended to demonstrate that the addition of the skydome quadrant weighting vectors w_i improved the accuracy of the sparse skydome feeler. As shown in Figure 4.11a, a simple scene consisting of an open-faced pipe oriented flush to a ground plane was constructed. The height of the pipe is equal to its inner wall radius. As such, the



Figure 4.11: Illustration of simple scene (a) and associated point cloud (b) for sparse skydome feeler validation.

percentage of hemisphere visible to the ground point at the center of pipe may be calculated as $e^{2\pi} e^{\pi/4} = e^{\pi/4}$

$$F = \frac{\int_0^{2\pi} \int_0^{\pi/4} \sin\theta \,\mathrm{d}\theta \mathrm{d}\phi}{2\pi} = 0.2929 \quad . \tag{4.7}$$

The sparse skydome feeler code was run on the point cloud shown in Figure 4.11b. Without the quadrant weighting factor, the output of the sparse skydome feeler code was 0.5 due to the fact that half of the sample directions had zenith angles that were greater than 45°. With the inclusion of the weighting factor, the output was equal to 0.2929. Obviously, the accuracy of the sparse skydome feeler output will be dictated by the point cloud's post spacing and the directional sampling density of the sparse skydome feeler itself; however, the experiment does demonstrate that the sparse skydome feeler theory is correct.

4.1.4 Preliminary Target Point Extraction

As defined in Eq. (1.2), the mixing fraction term M is used to quantify target abundance for a given pixel. Clearly, if one were able to predict M for every pixel in a hyperspectral image prior to analyzing the image's spectral data, the task of running a target detection algorithm on the spectral data set might seem redundant at best. However, if the accuracy in the prediction were in question, then additional processing would be warranted. Such is assumed to be the case when using Lidar-based shape information to estimate M. To be fair, this assumption may only be valid when the number of samples on target is less than 200 [Johnson 1997].

Obviously the task of estimating M from 3D point cloud data implies a search for spatial features associated with a target. If a target's shape is unknown at the time of search, then the process may be described as *anomaly detection*. That is, shape features that do not occur regularly in the scene background are characterized as anomalies and are subsequently identified as such to a user. In the event that a target's shape or other unique signature (*i.e.*, spectral reflectance) is known, then the search process may be described as *target detection*. When more than one object from a target class is present in a scene, then an additional step of target identification may need to be performed. The forthcoming discussion on using spin-image matching to estimate mixing fractions relates directly to the target detection process. Many aspects of the target detection process below are taken directly from Johnson's dissertation on using spin-image techniques for enhanced robotic vision [Johnson 1997]. Vasile then extended the spin-image technique to remotely sensed data sets and also demonstrated that Lidar-based target identification may be possible with sufficient resolution in the point cloud [Vasile and Marino 2004].

4.1.4.1 Z-windowing Above Ground Plane

One obvious disadvantage associated with performing an exhaustive search on a 3D point cloud for a known target shape is the long run time associated with doing so. By making an assumption about the orientation of targets relative to the ground plane, a significant number of points may be eliminated from subsequent shape detection algorithms. More specifically, the target is assumed to be oriented flush to the ground plane. For example, if the target of interest were a car, then one could assume that all 4 wheels are in contact with the ground plane. If this assumption is valid, then a threshold window relative to the ground plane may be applied to remove points which are not likely to be target samples. Applying a threshold relative to the ground plane is more complicated than simply applying a threshold to the z-dimension of the point cloud data. This is due to the fact that ground topology is rarely perfectly flat. More often, the ground plane exhibits a variety of local maxima and minima. These terrain artifacts may be minimized by *normalizing* the ground plane, as is illustrated in Figure 4.12.



Figure 4.12: Illustration of z-windowing. (a) Threshold on original data and (b) threshold after normalizing the ground plane.

Per Section 4.1.2, the output of many Lidar ground plane extraction algorithms is the subset of 3D points that is believed to be representative of the ground. This set of ground points forms an irregularly-gridded surface in 3D space. For the purpose of subtracting the ground plane from the point cloud data, the ground points should be interpolated using the (x, y) coordinates of the point cloud data as abscissa locations. Then the ground plane may be normalized by subtracting the interpolated ground point locations from the original point cloud data. Once in the normalized space, an upper and lower z-threshold may be applied in order to flag points that fall within the threshold boundaries. These flags should correspond to a point index in the original cloud. The original scene points corresponding to flagged indices are then retained. It is important to realize that the described flagging method does not alter the 3D geometry of the z-windowed points. Such may not be the case if points from the normalized space are retained as output.

Clearly, in order for this process to be robust, accurate extraction of the ground plane is critical. While increasing the range associated with the threshold window may ease the accuracy requirements on the ground plane extraction, it may come at the expense of adding unwanted points to the remaining point cloud. Ultimately, it is important to keep in mind that the process of z-windowing relative to the ground plane is an *optional* process. The subsequent spin-image based filtering techniques will still work, albeit with higher potential for false alarms and longer run times.

4.1.4.2 Spin-image Fundamentals

As stated earlier in Section 2.4, a spin-image $S_{\mathbf{p}}$ is a 2D, parameter space representation of 3D shape information relative to a single, oriented point \mathbf{p} [Johnson 1997]. This point will be referred to as the *spin-image basis point*. Using an eigenvector decomposition technique similar to that described in Section 4.1.2.2, the normal vector associated with \mathbf{p} may be estimated and is accordingly called the *spin-image basis normal* $\mathbf{n}_{\mathbf{p}}$. As illustrated in Figure



Figure 4.13: Illustration of spin-image parameters [Johnson 1997].

4.13, the first spin-image parameter α is defined as the perpendicular distance, relative to $\mathbf{n}_{\mathbf{p}}$, to point \mathbf{x} . The second parameter β is the signed, parallel distance, relative to $\mathbf{n}_{\mathbf{p}}$, to point \mathbf{x} . The mathematical representation of these parameters, also known as *spin coordinates*, is

$$\alpha = \sqrt{\|\mathbf{x} - \mathbf{p}\|^2 - (\mathbf{n}_{\mathbf{p}} \cdot (\mathbf{x} - \mathbf{p}))^2}$$

$$\beta = \mathbf{n}_{\mathbf{p}} \cdot (\mathbf{x} - \mathbf{p})$$
 (4.8)

Spin coordinates, relative to \mathbf{p} , may be calculated for every point \mathbf{x} in the 3D point cloud. To preserve accuracy in the spin-image generation process, spin coordinates are calculated using points from the raw point cloud. Notice that the spin coordinates (α , β) are measured in ungridded, spatial coordinates. The dimensionality reduction associated with going from 3D coordinates into the spin-image domain is described by

$$[x, y, z] = [r, \theta, z] \to [r, z] \approx [\alpha, \beta] \quad . \tag{4.9}$$

Assuming accurate normal estimation, the spin coordinates (α, β) should closely approximate the cylindrical coordinates (r, z), relative to point **p**. The cylindrical coordinate θ is ignored in the spin-image domain. A spin-image is simply a 2D histogram of the spin coordinates (α, β) associated with all points that meet three user-specified spin-image generation criteria C_s which will be described later.

The first step in building a spin image is determining which points may contribute to the spin image $S_{\mathbf{p}}$. This is accomplished by gridding the 3D data in spin parameter space and applying the spin-image generation criteria to the list of gridded points $L\{\mathbf{x}\}_{\mathbf{p}}$. Based on the (α, β) spin coordinates associated with all of the points in $L\{\mathbf{x}\}_{\mathbf{p}}$, the data is gridded according to

$$i = \left\lfloor \frac{\alpha}{b_s} \right\rfloor$$

$$j = \left\lfloor \frac{\frac{W}{2} + \beta}{b_s} \right\rfloor$$
(4.10)

At this point, the first two spin-image generation criteria are introduced: spin support W and bin size b_s . The first criterion W is used to specify the range, relative to \mathbf{p} , over which points are allowed to contribute to a spin-image. Much like any histogram, b_s specifies the range over which spin coordinates will map to a single bin. In practice, the bin size is typically set to four times the mean sampling resolution associated with the point cloud. This is done to ensure that multiple points may map to the same bin, therein creating spin-image contrast features. These contrast features are important when performing spin-image matching, as described in Section 4.1.4.3. Only points from $L\{\mathbf{x}\}_{\mathbf{p}}$ that meet the following conditions

$$i < \left\lfloor \frac{W}{b_s} \right\rfloor$$

$$j < \left\lfloor \frac{W}{b_s} \right\rfloor$$
(4.11)

may contribute to $S_{\mathbf{p}}$. The final spin-image generation criterion, spin angle Λ_x , is defined as

$$\Lambda_x = \cos^{-1} \left(\mathbf{n}_{\mathbf{p}} \cdot \mathbf{n}_{\mathbf{x}} \right) \quad . \tag{4.12}$$

 Λ_x is a measure of the angle between $\mathbf{n_p}$ and the normal vector $\mathbf{n_x}$ associated with point \mathbf{x} . Λ_x is calculated for every point in $L\{\mathbf{x}\}_{\mathbf{p}}$ meeting both conditions specified in Eq. (4.11). Then, only the points with spin angles that are less than a user-specified threshold Λ_s are saved for the final list of available points $L\{\mathbf{x} \cap C_s\}_{\mathbf{p}}$.

Once $L\{\mathbf{x} \cap C_s\}_{\mathbf{p}}$ has been established, a bilinear interpolation is used to reduce the error associated with the gridding process. Notice that the gridding error introduced by the floor operators in Eq. (4.10) is directly proportional to b_s . This error is illustrated in Figure 4.14 below. The green dot represents a point mapped according to its continuous spin coordinates (α, β) into spin space, while the red dot shows the result of gridding using the floor operator. To mitigate this error, the spin coordinates (α, β) and gridding indices (i,j) are used in a bilinear interpolation to calculate the contribution of a point to its surrounding grid locations, indicated by the blue dots. Bilinear interpolation weights are calculated



Figure 4.14: Illustration of gridding error and bilinear interpolation.

according to

$$a = \alpha - ib_s$$

$$b = \beta + \frac{W}{2} - jb_s$$

$$(4.13)$$

A point's contribution to $S_{\mathbf{p}}$ resulting from the bilinear interpolation is calculated via

$$S_{\mathbf{p}}(i, j+1) = S_{\mathbf{p}}(i, j+1) + a(b_s - b) \quad S_{\mathbf{p}}(i+1, j+1) = S_{\mathbf{p}}(i+1, j+1) + (ab)$$

$$S_{\mathbf{p}}(i, j) = S_{\mathbf{p}}(i, j) + (b_s - a)(b_s - b) \quad S_{\mathbf{p}}(i+1, j) = S_{\mathbf{p}}(i+1, j) + b(b_s - a)$$

$$(4.14)$$

Finally, all 3D points in $L\{\mathbf{x} \cap C_s\}_{\mathbf{p}}$ are binned according to Eq. (4.14) to form $S_{\mathbf{p}}$. It is important to recognize that a spin-image may be generated for every point in a 3D point cloud. Figure 4.15 illustrates three example spin images generated from a 3D-meshed rubber duckie model [Johnson 1997]. Each image pair, outlined in the colored boxes, is a point-specific, spin domain representation of the continuous (α, β) mapping of the model and the resulting gridded version of the same data, post bilinear interpolation. A spin-image representation of a 3D object may or may not be immediately recognizable, as spin-image



Figure 4.15: Example spin images from rubber duckie [Johnson 1997].

features are directly related to radial symmetries about **p**. Since (α, β) coordinates are measured relative to **p** and **n**_p, spin-images are invariant to object orientation.

4.1.4.3 Spin-image Target Detection

The task of target detection implies (1) a user has a data set where one or more instances of a single, known target may or may not be present and (2) a user knows a unique target signature which may, in part or in whole, be present in the data set. For the scenario of spinimage target detection, the necessary target signature is obviously the target's 3D shape. Spin-image target detection fundamentally entails matching scene-based spin-images to a spin-image library built from a 3D model of a target. A scene-based spin-image S_s associated with point s is compared to every spin-image in the target library. The best match between S_s and the spin-image library is referred to as a *correspondence* $C(S_s, S_m)$, where S_m is the matched, model-based spin-image from the library [Johnson 1997]. Correspondences are then sorted according to their respective matching criterion scores and filtered based on expected geometric relationships. Finally, assuming the spin-image target detection process is accurate, the remaining correspondences indicate instances of where point \mathbf{m} from the model appears at point \mathbf{s} in the scene. A block diagram of the spin-image target detection process is provided in Figure 4.16.



Figure 4.16: Block diagram of spin image target detection process.

Before addressing the specifics of spin-image target detection, a relevant question is introduced. Should one expect a spin-image that is generated from an instance of a target within a scene to perfectly match a spin-image from the target library? Intuition should tell the reader that the answer is most certainly no; however, the answer to the inevitable question, "Why not?" may not be as immediately apparent. First and foremost, the same spin-image generation criteria must be used when building S_s and S_m . This also implies that instances of the target within a scene must be roughly the same scale as the target model. As such, spin-images are not scale invariant. Knowledge of the Lidar sensor's location and the extracted ground plane should allow for sufficient scaling of the model. Secondly, occlusion may be present in the scene. Two forms of occlusion are considered: external scene occlusion and self-occlusion. The former occurs when background artifacts block a sensor's view of part of a target, while the latter occurs due to illumination of a target from a single direction, also known as *self-shadowing*. When points are missing from a target instance, the associated spin-image bins where the missing points would have mapped have reduced (or no) population. Finally, the spatial sampling resolution of the 3D target model may be different from that of the Lidar sensor. Often, the target model is sampled at a much finer resolution that what is feasible from a topographic Lidar sensor. Such a scenario might result in loss of fine-scale spatial features in the scene-based spin-image. The following sections address methods for dealing with these issues, with the overall objective of building a robust target detection scheme.

Library Generation

Spin-image library generation is among the first steps in spin-image target detection. As discussed in Section 4.1.4.2, spin-images are unique, 2D representations of the shape information associated with 3D objects or scenes. In order to build a spin-image library, one must first have a 3D spatial model of the target. This model should be unoccluded, appropriately scaled to match the expected target size, and of sufficient resolution to capture the 3D structure of the target. Next, the model should be spatially sampled at regular intervals to arrive at a noise-free, 3D point cloud. Again, the spatial sampling interval should be at fine enough resolution to fully characterize the 3D nature of the target. Often, this spatial sampling interval will be much smaller than what may be expected in the scene data. Finally, a spin-image should be created for every point in the 3D point cloud.

In practice, there are a variety of ways to generate a point cloud from a 3D object model. Most models, such as 3D Studio Max^{TM} and *obj*-based files, are facetized versions of a continuous object. While one could use the vertex information associated with these facet-based models as a point cloud, doing so may be problematic as the vertices are not typically spaced at regular intervals. For example, a single rectangular facet might be used to model the hood of a vehicle. In such a case if only the hood's vertex coordinates were used, much of the planar information associated with the hood would be lost and therefore not contribute to the vehicle's related spin-images. The most practical way to deal with this issue is to implement a fixed interval ray tracer from a variety of viewpoints about the model. A point should be added to the model point cloud at each intersection of a ray and continuous model geometry. The view points should be varied such that all sides of the object are sampled in order to avoid self-shadowing effects.

To generate a library of spin-images, every point in a model point cloud is used as a spin-image basis point. Hence, a cloud consisting of *n* points results in a library of *n* images. Such a library may be considered to be a 3D data cube. While library generation may seem relatively straightforward, the subtleties of intelligent library generation lie in the selection of the spin-image generation criteria. The first criterion, bin size, should be sufficiently large to allow multiple points to map to a single bin. It is important to consider the *contrast versus resolution* tradeoff when picking the bin size. While increasing the bin size often increases the contrast across a spin-image, it also degrades the resolution in the spin-image. Considering that the same spin-image generation criteria are applied to both the model and the scene, the scene point cloud's spacing interval should drive the bin size criterion. As such, the bin size should be set to one to two times the scene's mean post spacing.

A logical choice for the second criterion, bin support, is the length of the object along its major axis. Figure 4.17 illustrates an M1 tank with a dashed red line marking the vehicle's major axis. Using this construct for selecting W is ideal if one expects a scene-based target to appear with little to no external occlusion present. However, due to the fact that the largest logical size for W is being used, the corresponding spin-image is also maximized in size, thereby increasing the computational expense associated with spin-image matching. If a user expects a target to be partially hidden, then using a smaller length for W may be warranted, with the additional benefit of shorter processing times.



Figure 4.17: Major axis along M1 tank.

The final spin-image generation criterion, spin angle, is useful for building self-occlusion effects into a model. The spin angle criterion should *not* be applied when generating scene-based spin-images, as self-occlusion is inherent in the data set. While general rules-of-thumb may be applied for setting the previous two generation criteria, picking an appropriate value

for the spin angle threshold Λ_s is target-dependent and may be somewhat subjective. While a logical upper limit for Λ_s is 180°, using a smaller angle such as 90° may be useful for objects characterized by large-scale corner shapes. For example, Figure 4.18 represents a 3D cube and its associated surface normal vectors. Notice that only the surfaces with normals that are within 90° of a particular normal are viewable, regardless of cube orientation and/or viewing geometry. If one were to build a spin-image library for the cube, using a spin angle limit of 90° would prevent non-viewable surface points from one side of the cube from contributing to the model-based spin-images. If the same cube were oriented within a scene such that only one face were viewable, its associated spin-images would not be well-matched to a spin-image library built from a model with points from five sides of a cube. This example illustrates



Figure 4.18: 3D cube and its surface normal vectors.

how the spin angle criterion does incorporate some self-occlusion effects into a spin-image library; however it is not a rigorous characterization. Furthermore, picking a logical Λ_s value for real-world targets may not be as straightforward as the cube example above. In many cases, a target is modeled as a combination of many different primitive object shapes, such as cubes, cylinders, and conic sections. As such, a user should carefully consider a target's dominant shape features when selecting this threshold value.

Matching Criterion

Once the model-based library has been established, a method for matching a scene-based spin-image S_s to a spin-image from the library is needed. A criterion \mathbb{M} for comparing the quality of a match between a given spin-image S_s and the various spin-images in the model-based library is defined as

$$\mathbb{M} = \mathbf{r}(S^*_{\mathbf{s}}, S^*_{\mathbf{m}}) \cdot N \quad . \tag{4.15}$$

The correlation coefficient r, ranging between 1 and -1, measures the degree to which the function's arguments are linearly related. If a target appears in a scene, one might expect the spin-images associated with target scene points to match closely to one or more spin-images from the model-based library. Additionally, the correlation coefficient between quality correspondence candidates should be close to 1. However, if a target is occluded in the scene, the correlation coefficient will degrade as the occlusion level increases. In order to prevent such degradation in r, only filled-bins which are common to both the scene- and model-based spin-images should be compared. Hence, the arguments of r are the scene/model spin-images where overlap, annotated by *, occurs. N is the number of non-zero bins that are present in S_s^* and S_m^* . The N term is important because it weights the overlapped correlation coefficient by the amount of overlap. Hence, M for a spin-image pair with only a few overlapped, highly-correlated bins may be lower than that for a spin-image pair with many overlapped, moderately-correlated bins. Finally, the correspondence for scene point

s is defined as the model spin basis point **s** that results in the highest \mathbb{M} score. It should be noted that the metric \mathbb{M} is very similar to Johnson's *similarity* metric *S*; however, a primary difference between the two is that Johnson's metric requires an *a priori* estimate of the amount of target present in a scene [Johnson 1997].

Calculation of the M score provides an initial framework for filtering out bad scene/model correspondences. Once a model point correspondence has been established for every point in the scene, the correspondences are sorted according to their respective M scores. All correspondences with an M score that is greater than or equal to one half the maximum M score are retained [Johnson 1997].

Geometric Consistency

After the preliminary M-based filtering, the remaining, collective correspondence set is queried for geometric consistency. This geometric consistency filter is implemented in the spin domain and ranks point correspondences using (1) the relative distance between the correspondence's spin-image basis points and (2) the orientations of the correspondence's spin-image basis point normals. Recall that a correspondence consists of a scene-based spinimage and the best matched model-based spin-image, both of which are generated relative to their respective spin-image basis points. A point correspondence $P(\mathbf{s}, \mathbf{m})$ is analogous, as it defines the scene/model spin-image basis points associated with the correspondence's respective scene/model spin-images. A point correspondence is generated for every point in the scene. The fundamental premise driving the geometric consistency filter is that the spatial relationships between point correspondences in the model domain should be similar to those in the image domain. Point correspondence pairs that are spatially inconsistent with the entire point correspondence set should be removed. Figure 4.19 illustrates the concept of geometric consistency. The aqua field in Figure 4.19(a) contains two model-domain point



Figure 4.19: Illustration of geometric consistency concept [Johnson 1997].

correspondences, while the pink field in Figure 4.19(b) shows the corresponding scene-domain point correspondences. The green lines between the respective domains' points illustrate the spin coordinates of $\mathbf{s}_2/\mathbf{m}_2$ relative to the first point correspondence $P_1 = P(\mathbf{s}_1, \mathbf{m}_1)$. Likewise, the blue lines represent the spin coordinates of $\mathbf{s}_1/\mathbf{m}_1$ relative to the second point correspondence $P_2 = P(\mathbf{s}_2, \mathbf{m}_2)$. Notice that the lengths of the green line segments are relatively the same in both domains, indicating good geometric consistency. The same also applies to the blue line segments. Consistency in the line segments depends not only on the relative distance between the point correspondences, but also on the normal vector orientations at each point. The figure's point correspondences have similar spatial separation and relative normal vector orientation in both domains. As such, these point correspondence pairs would pass the geometric consistency filter. The geometric consistency of two point correspondence pairs is calculated according to

$$d_{gc}(P_1, P_2) = \frac{\|\mathbf{S}_{\mathbf{m}_2}(\mathbf{m}_1) - \mathbf{S}_{\mathbf{s}_2}(\mathbf{s}_1)\|}{(\|\mathbf{S}_{\mathbf{m}_2}(\mathbf{m}_1)\| + \|\mathbf{S}_{\mathbf{m}_2}(\mathbf{m}_1)\|)/2} \qquad .$$
(4.16)
$$D_{gc} = \max(d_{gc}(P_1, P_2), d_{gc}(P_2, P_1))$$

 $d_{gc}(P_1, P_2)$ is a quasi-normalized, scalar metric that describes the difference in the distance between the model-domain point correspondences $\mathbf{m}_1/\mathbf{m}_2$ and the scene-domain point correspondences $\mathbf{s}_1/\mathbf{s}_2$, as measured in spin space. The order of the input arguments to d_{gc} is not trivial. The first argument P_1 indicates that the spin coordinates to $\mathbf{s}_1/\mathbf{m}_1$ will be calculated. The second argument P_2 specifies the spin-image basis point location from which the measurement to s_1/m_1 will be made. Hence, $S_{m_2}(m_1)$ is a 2D vector consisting of the (α,β) coordinates to point \mathbf{m}_1 , as measured from point \mathbf{m}_2 . The denominator is the average magnitude of the numerator terms. A point correspondence with good geometric consistency should have a low value for d_{gc} . The denominator term is necessary to prevent a bias in d_{gc} toward point correspondence pairs that are in close proximity to one another. D_{gc} provides a means for assessing the consistency in the relative orientations of the point normals associated with the point correspondence pairs. When D_{gc} is small then the point correspondence pairs P_1 , P_2 are geometrically consistent. However, if D_{gc} is a high value, then either one or both of the point correspondences could be geometrically inconsistent. Therefore, further steps must be taken in order to filter out the geometrically inconsistent points. The following pseudocode describes how the geometric consistency filter is implemented.

- \bullet ; begin with a scene consisting of m points
- FOR i = 1, m DO BEGIN

- Create scene spin-image S_{i} for point i
- Determine best library match to $S_{\mathbf{i}}$ according to $\mathbb M$
- Store model point correspondence for i and match score ${\mathbb M}$
- END FOR
- Remove all point correspondences with $\mathbb{M} < \frac{1}{2}\max(\mathbb{M})$
- ; After filtering based on $\mathbb M$, a list of n point correspondences remains
- FOR j = 1, n DO BEGIN
 - Select P_j (comprised of a model point ${f m}$ and scene point ${f s}$)
 - Calculate distance from s to all scene point correspondences
 - Store q scene correspondences with distances (relative to s) that are less than the spin support criterion
 - IF q > 4 THEN BEGIN
 - * Initialize below threshold count = 0
 - * FOR k = 1, q DO BEGIN
 - \cdot Compute $d_{gc}(P_j,P_k)$
 - · Compute $d_{gc}(P_k, P_j)$
 - \cdot Compute D_{gc}

· IF $D_{gc} < T_{gc}$ THEN below threshold count = below threshold count +1

* END FOR

* IF below threshold count $> \frac{1}{4}n$ then begin

 \cdot P_j passes geometric consistency filter

 \cdot Store P_j and its associated $below\ threshold\ count\$ score

- * END IF
- END IF ELSE BEGIN
 - * P_j fails geometric consistency filter
 - * Store P_j and its associated below threshold count score
- END ELSE
- END FOR

The primary component of the pseudocode is a double FOR-loop (highlighted in red), which allows for an exhaustive geometric consistency check. T_{gc} is a user-defined threshold that determines if a single pair of point correspondences is geometrically consistent. Often, a value of 0.25 is used for T_{gc} in order to enforce strong geometric consistency. Notice that each point correspondence must be geometrically consistent with at least $\frac{1}{4}$ of the possible correspondences in order to pass the geometric consistency filter [Johnson 1997]. If so desired, a user may make the geometric consistency filter more stringent by decreasing T_{gc} . A final set of spin-image pairs should be generated from the point correspondences that passed the geometric consistency filter. Section 4.1.4.3 discusses how this set of spin-images may be used to estimate which points in a Lidar point cloud correspond to target samples.

Recovery of 3D Target Points

Before addressing the final target point extraction method, it is important to re-emphasize what comes out of the traditional spin-image process. Prior to processing a scene, a spinimage library is generated for a single 3D model – that is, a spin-image is created for every point in the 3D model. During spin-image filtering of a scene, a single spin-image is created for every point in the scene. Additionally, every point in the scene is assigned its best model point match, based on computing a similarity score for every spin-image in the model library relative to the scene point's spin-image. A scene point and its best model point match forms a correspondence pair. All correspondence pairs are then filtered according to (1) the strength of the similarity matching score and (2) geometric consistency. The point correspondences that pass both filters are deemed *valid correspondences*. Figure 4.20 is a



Figure 4.20: Three notional correspondence points between a model (a) and scene (b).

conceptual illustration of correspondence pairs, with the continuous geometry for the model and scene shown for clarity. Note that the correspondences are color-coded, corresponding to match points between the scene and model. The blue and green correspondences appear to be valid, while the geometric consistency check should eliminate the red correspondence pair. Although a spin-image inherently represents a reduction in the dimensionality of 3D spatial data, it does not necessarily mean that the 3D information is discarded. For example, if one were to assign an index to every point in a scene cloud, then the indices for points that contributed to a particular spin-image could be stored. Furthermore, scene points contributing to the overlapped bins between a valid correspondence pair's spin-images could also be stored. These scene points are likely to include many target points; however, some background points may map into the overlapped bins and also be included. Admittedly, this process of extracting target points is likely to have a significant number of false alarm points; however, subsequent spectral processing of products based on this preliminary target cue is likely to suppress false alarms. As such, the final output of the target point detection process is achieved by using the above-described method to extract target points from all valid correspondence pairs.

While the scene-points from the valid correspondences alone could be used as a target point output, it may not be reasonable to assume that every point on a target within the scene will yield a valid correspondence pair. The strength of the proposed target point extraction process is that only a single valid correspondence on or even close to (*i.e.*, within the spin-image support criterion) a target is necessary to subsequently extract all point samples from the target.

It is important to keep in mind that the accuracy of the spin-image target detection process may be questionable. The list of valid correspondence pairs for a scene may contain false alarm correspondences, scene points that do not truly correspond to a point on the model. Vasile and Johnson report similar numbers on the order of 200 for the required number of samples on target to ensure robust detection performance [Vasile and Marino 2004; Johnson 1997]. For most remote sensing Lidar sensors, these sampling densities are not feasible for all but the largest of targets. As such, additional spectral processing is warranted and necessary.

4.1.5 Sparse Z-buffer

Section 4.2 describes a variety of 2D features maps which are derived by projecting the 3D point cloud and its associated features onto the spectral focal plane array. In the event that the Lidar and spectral sensors are not located on the same platform, then an additional preprocessing step *prior to* the point cloud processing techniques described in Sections 4.1.1-4.1.4 may be required. Every point in the cloud should be labeled according to whether or not it is visible to the spectral sensor. Only points that are visible should be processed for the various map features. This preprocessing step is referred to as a *sparse z-buffer*. Traditional *z*-buffering is common in computer graphics and is useful for 3D visualization of continuous facets relative to a single viewing location. Again, since Lidar data is inherently non-continuous, a sparse implementation, based on the same logic as the sparse shadow feeler described in Section 4.1.1, is used. A ray is traced from a scene point under test to the spectral sensor location. If a cylinder/sphere intersection occurs with another point in the scene in order to ensure that the subsequent feature maps are accurate relative to the spectral sensor.

Point Cloud Processing Summary

All of the methods introduced in Sections 4.1.1-4.1.5 are designed to process Lidar data directly. While many of the methods are based on traditional techniques developed by the computer graphics community, the suggested methods present an attractive alternative due to their ability to handle irregularly gridded data sets. Each of the processes assigns one or more feature labels to points within the cloud, without altering the locations of the points. In the event that a gridded product is required by the user, a methodology for transforming a point cloud and its respective feature labels into a user-defined gridded reference frame is presented in Section 4.2.

4.2 Map Creation

4.2.1 3D Projection onto HSI FPA

In order for feature points, as identified by Lidar processing, to be transformed into a feature map relative to the HSI FPA, the points must be projected from their native 3D locations onto the 2D focal plane. This task is necessary because the GIFOV of a hyperspectral sensor is dependent upon the range from the sensor to the scene. Put differently, the projected area of a hyperspectral pixel will vary with spatial variation in the range dimension of the scene. A framework is developed in Eqs. (4.17-4.24) to transform the 3D spatial coordinates of a Lidar point into indexed pixel coordinates associated with the hyperspectral FPA. This transformation involves two basic steps. First, one must determine the continuous location to which a scene point maps. Second, this 3D spatial location must be converted

into a 2D, indexed pixel coordinate.

A reference frame may be defined as a set of three orthogonal unit vectors, specifying positive directions for locating one or more points in 3D space. As mentioned earlier in Section 4.1.2, UTM coordinates provide a common reference frame for geospatial data sets. Also, a sensor typically has its own reference frame that is unique to the sensor's pointing geometry. Obviously, two of the unit vectors will lie in the same plane as the sensor's FPA, while the third identifies the FPA normal vector. Eq. (4.17) is a notional reference frame for a sensor with a nadir viewing geometry. Notice that the third column indicates that the sensor is pointing downward. Implicit in this construction is the assumption that the sensor's reference frame is defined relative to a global coordinate system.

$$S = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \quad . \tag{4.17}$$

The orientation of a sensor's reference frame may be altered by applying a 3D rotation matrix M_{xyz} defined as

$$M_{xyz} = \begin{bmatrix} \cos(\theta_z)\cos(\theta_y) & \cos(\theta_z)\sin(\theta_y)\sin(\theta_x) - \sin(\theta_z)\cos(\theta_x) & \cos(\theta_z)\sin(\theta_y)\cos(\theta_x) + \sin(\theta_z)\sin(\theta_x) \\ \sin(\theta_z)\cos(\theta_y) & \sin(\theta_z)\sin(\theta_y)\sin(\theta_x) + \cos(\theta_z)\cos(\theta_x) & \sin(\theta_z)\sin(\theta_y)\cos(\theta_x) - \cos(\theta_z)\sin(\theta_x) \\ -\sin(\theta_y) & \cos(\theta_y)\sin(\theta_x) & \cos(\theta_y)\cos(\theta_x) \end{bmatrix} .$$

$$(4.18)$$

 θ_x , θ_y , and θ_z define a 3D angular rotation about the sensor's local coordinate system x, y, and z axes, respectively [Brown 2005]. M_{xyz} may be broken down into three separate 2D rotations (*i.e.*, M_x , M_y , and M_z) about these axes. As such, it should be noted that the form of M_{xyz} only holds for a specific ordering of 2D rotations.

A generalized definition of a sensor reference frame within a scene is given by

$$P = M_{xyz}S \quad . \tag{4.19}$$

The columns of P are unit vectors specifying the sensor's pointing direction and orientation. Unfortunately, when hyperspectral scene collects are made, the angles θ_x , θ_y , and θ_z are often unknown. However, this may not be an issue, as a sensor's location and pointing direction are outputs from a Global Positioning System (GPS) and Inertial Navigation System (INS), both of which are commonly found on commercial sensors. Hence, in real-world applications, the sensor's reference frame P for a given collection geometry is often known to within the error of the GPS/INS.

Given, a sensor's position \mathbf{s} , orientation, and effective focal length f, one may solve for the planar intersection of any point \mathbf{g} in the scene according to [Schneider and Eberly 2003]

$$\mathbf{P}_{gf} = \frac{\mathbf{s} + f\mathbf{P}_3 - \mathbf{g}}{\|\mathbf{s} + f\mathbf{P}_3 - \mathbf{g}\|}$$
$$t = \frac{-(\mathbf{P}_3 \cdot \mathbf{g} - \mathbf{P}_3 \cdot \mathbf{s})}{\mathbf{P}_3 \cdot \mathbf{P}_{gf}} \qquad (4.20)$$
$$\mathbf{q} = \mathbf{g} + t\mathbf{P}_{qf}$$

 \mathbf{P}_i is a unit vector specified by the *i*-th column of *P*. \mathbf{P}_{gf} is defined as the unit vector pointing from scene point **g** to the sensor focal point. *t* is the scalar length of a line segment from **g** to its corresponding point of intersection on the FPA. Hence, *t* may be used to solve for the intersection point \mathbf{q} on the FPA. This construction is illustrated in Figure 4.21.



Figure 4.21: Illustration of vector geometry for planar projection.

The second step involved in transforming the 3D spatial coordinates associated with point \mathbf{g} is determining the FPA array bin to which the point maps. Figure 4.22 illustrates the geometry associated with this transformation. Because the sensor reference frame and all 3D point coordinates are defined relative to the global coordinate system, converting \mathbf{q} from 3D spatial coordinates to indexed pixel coordinates for the FPA is relatively simple. First, the vector from \mathbf{s} to \mathbf{q} is computed as

$$\mathbf{R}_{sq} = \mathbf{q} - \mathbf{s} \quad . \tag{4.21}$$

Naturally, the magnitude of \mathbf{R}_{sq} is the radial distance from \mathbf{s} to \mathbf{q} . Since both \mathbf{s} and \mathbf{q} lie



Figure 4.22: Converting **q** from spatial to indexed coordinates.

in the plane defined by P_1 and P_2 , the linear projection of \mathbf{R}_{sq} onto the sensor's respective in-plane reference frame vectors,

$$u = \mathbf{R}_{sq} \cdot \frac{\mathbf{P}_1}{\|\mathbf{P}_1\|} + o_x$$

$$v = \mathbf{R}_{sq} \cdot \frac{\mathbf{P}_2}{\|\mathbf{P}_2\|} + o_y$$
(4.22)

yields the 2D spatial position (u, v) of **q** relative to **s**, as measured by the sensor's reference frame. Note that o_x and o_y , defined as

$$o_x = \frac{L_x}{2} , \qquad (4.23)$$
$$o_y = \frac{L_y}{2}$$

specify the center of the FPA. L_x and L_y are the lengths associated with the photo-active area of the entire FPA in the horizontal and vertical dimensions, respectively.

Finally, the continuous, spatial coordinates (u, v) may be converted to indexed coordinates (i, j) specific to the FPA according to

$$i = \left\lfloor \frac{u}{b_x} \right\rfloor$$

$$j = \left\lfloor \frac{v}{b_y} \right\rfloor$$
(4.24)

The terms b_x and b_y are the lengths along the sides of a single FPA pixel, commonly known as pixel pitch. It should be noted that this framework does not account for geometric distortion in the sensor's optics, nor does it account for distortion due to scanning mechanisms or spectral artifacts. While the above derivation applies directly to mapping 3D points onto a 2D FPA used in framing HSI systems, a similar approach could be used to model linear arrays used in pushbroom scanning HSI systems. In such a case, the HSI sensor's origin **s** and reference frame P would need to be updated as the sensor's field of view (FOV) traverses the sensor ground track.

4.2.2 Shadow Map - *K*

The framework developed in Eqs. (4.17-4.24) applies directly to the creation of all feature maps. However, the exact implementation of this projection varies slightly based on the feature map in question. For the case of creating a shadow map, 3D points that have been labeled as being in shadow via the sparse shadow feeler algorithm are projected onto a finer grid than that of the HSI FPA, as is illustrated in Figure 4.23. The bold grid lines represent the coarse grid associated with the HSI FPA, while the fine grid lines indicate the fine-scale subgrid. Gray pixels show the subgrid location to which shadowed points mapped. The



Figure 4.23: Illustration of shadow projected onto subgrid.

fidelity of the fine-scale subgrid allows for the assessment of an HSI pixel's shadow purity – that is, how much of an HSI pixel is occupied by shadow. Shadow purity K for an HSI pixel is calculated as

$$K = 1 - \frac{\sum_{i=1}^{n} k_i}{n^2} \quad , \tag{4.25}$$

where k_i is binary fill value for the *i*-th subgrid location within the HSI pixel and *n* is the number of locations in the subgrid along a single dimension. Hence, the HSI pixel in the upper left corner of Figure 4.23 has a *K* value of 7/16 or 0.4375. In practice, *n* should be set equal to the amount of spatial oversampling expected relative to the HSI GIFOV. This oversampling value may be calculated as

$$n = \left(\frac{GIFOV}{PS}\right)^2 \quad , \tag{4.26}$$

where GIFOV represents the ground instantaneous field of view of the HSI sensor and PS is the mean post spacing of the Lidar point cloud. In the event that the Lidar point cloud has different mean post spacings in the along-track and scan directions, as is often the case for line scanned systems, then the calculation of n changes simply to

$$n = \left(\frac{GIFOV^2}{PS_x PS_y}\right) \quad . \tag{4.27}$$

Given the fact that a single shadow point maps to a single subgrid location, using a subgrid that is finer than the oversampling value is likely to result in low purity estimates. Finally, calculating K for every HSI pixel location yields a grayscale shadow map image with values ranging from 0 to 1, with a value of 0 representing a pure shadowed pixel. These values represent local estimates of K as used in Eq. (1.2).

4.2.3 Illumination Angle Map - θ

As indicated in Eq. (1.2), accurate knowledge of the solar illumination angle relative to a local surface normal is important when modeling the direct solar loading upon the surface. Without knowledge of 3D scene geometry, orientation of a local surface normal is unknown. Previous methods for addressing this problem include, (1) assume the ground plane is flat and substitute the solar zenith angle for θ , (2) globally modulate the $\cos \theta$ term from 0-1 when generating a target subspace, (3) estimate local normal orientations from a DEM, or (4) estimate local normal orientations via a least-squares subspace approach, per Section 2.3.2.2. Given the availability of geo-registered Lidar and HSI data, the proposed method may prove to be more accurate.
Before the extracted ground plane from the Lidar data may be used to estimate θ , the ground plane points must be projected onto the hyperspectral focal plane. Eqs. (4.17-4.24) describe the transformation between the 3D point coordinates and indexed pixel coordinates on the HSI FPA. Unlike the creation of a shadow map described in the previous section, a fine-scale subgrid need not be defined when creating an illumination map. This is because all sub-pixel portions of an HSI pixel are likely to be populated by projected points.

The final step in the estimation of θ involves calculating the angle between the solar pointing vector \mathbf{p}_s and the integrated local point normal for a given hyperspectral pixel. The solar pointing vector is defined as the unit vector from the global coordinate system origin to the sun and may be derived from solar zenith and azimuth angles. Since many 3D Lidar terrain points may, and probably will, map to a single hyperspectral pixel, a mean point normal vector $\bar{\mathbf{n}}_{i,j}$ must be calculated for the hyperspectral pixel (i, j). Calculation of the angle between $\bar{\mathbf{n}}_{i,j}$ and \mathbf{p}_s is accomplished according to

$$\theta_{i,j} = \cos^{-1} \left(\frac{\bar{\mathbf{n}}_{i,j} \cdot \mathbf{p}_s}{\|\bar{\mathbf{n}}_{i,j}\| \|\mathbf{p}_s\|} \right) \quad .$$
(4.28)

Calculation of $\theta_{i,j}$ for every pixel in the HSI array yields a grayscale illumination map with values ranging from 0° to 90°.

The proposed estimation of θ is based on local normal vectors which are derived from *ground plane* point distributions. Building a target space using this approach implicitly assumes that a target material is parallel to the ground plane. Obviously, this assumption will not always be true. As an alternative, the user may also calculate θ using normal vectors from the original, raw point cloud. However, for low density point clouds, spatial sampling of

the underlying scene geometry may not be adequate to produce accurate estimates of surface normals, particularly in the case of complex scene features such as corners. Since an extracted ground terrain typically lacks such rapidly varying geometry, ground-derived normal vectors may be less susceptible to errors due to under-sampling and ultimately produce a more accurate estimation of θ .

4.2.4 Skydome Map - F

It is important to recognize that a rigorous estimation of the downwelled radiance contribution for a single pixel is *not* solely dependent upon geometric features that are measurable by a topographic Lidar system. This downwelled contribution not only depends on the integrated area of the exposed sky above a target (which is predictable using the sparse skydome feeler technique described in Section 4.1.3), but also on the distribution of the downwelled radiance throughout the hemisphere above a target. This downwelled distribution can vary dramatically with time of day, atmospheric composition, and cloud positions. Figure 4.24 shows an example downwelled distribution from a fully-exposed target [Schott 1997]. If tall, opaque features were placed immediately adjacent to the target, topographic Lidar information of the scene might provide enough information to predict the quadrants where the distribution would be suppressed. However, if a spatially-distributed cloud cover were added to the scene, topographic Lidar data would provide little insight as to how the downwelled radiance contribution might have changed. Hence, constraining a signature space using an F estimate derived from a 3D Lidar data set may not be readily feasible. That said, if one were to assume a clear sky, then the hemispherical distribution of downwelled radiance may be well characterized based on the sun's position in the sky. If a Lidar data set were collected



Figure 4.24: Integrated downwelled radiance distribution across the visible and near infrared spectrum.

from a near-nadir viewing geometry, then sparse skydome feeler ray tracing techniques could be applied to the 3D point cloud as a means to approximate the attenuation of the downwelled radiance distribution at each HSI pixel location. Note that the proposed method for estimating F for a given scene location, per Section 4.1.3, further simplifies the problem by assuming a uniform downwelled radiance distribution across the skydome, according to

$$F = 1 - \frac{\sum_{i=1}^{n} f_i w_i L_i}{\sum_{i=1}^{n} w_i L_i} \approx 1 - \frac{\sum_{i=1}^{n} f_i w_i}{\sum_{i=1}^{n} w_i} \quad .$$
(4.29)

Recall that f_i is the binary output from the sparse skydome feeler for the *i*-th sample direction in the skydome, while w_i is a weighting factor based on the solid angle subtended by the *i*-th sample direction quadrant. L_i is the spectrally integrated downwelled radiance distribution for the *i*-th sample direction. Clearly, the assumption of a uniform downwelled

radiance distribution is likely to be incorrect; however, the resulting estimate for the shape factor, and ultimately the downwelled contribution to the overall radiance for a pixel, may prove to be adequate if the downwelled contribution is small relative to the direct solar and upwelled radiance path contributions. Such is commonly the case for data collected under clear atmospheric conditions.

Once the Lidar point cloud has been processed by the sparse skydome feeler, then every point will have a value for F assigned to it. Each scene point then is projected onto the HSI focal plane via Eqs. (4.17-4.24). n scene points map to a single pixel location in the focal plane. The value for skydome map pixel (i, j) is calculated as the average of the F values for n points that map to pixel (i, j). The final skydome map is a grayscale image with values ranging from 0 to 1, which represent local estimates for F in Eq. (1.2).

4.2.5 Pixel Purity Map - M

A pixel purity map is a grayscale image used to represent target locations. Gray levels in the map are used to represent how much of the projected area of given pixel is filled by target material. As such, a value of 1 represents a pure target pixel. Per Section 4.1.4.3, the spin-image filtering process labels every point in the scene point cloud with a binary label indicating whether or not the point is believed to be a target sample. To create the purity map, all target samples are projected into the HSI FPA according to Eqs. (4.17-4.24). Since subpixel information is to be captured in the resulting pixel purity map, the points need be projected onto a fine-scale grid, much like in the creation of a shadow map (see Section 4.2.2). Finally, pixel purity is calculated as

$$M = \frac{\sum_{i=1}^{n^2} m_i}{n^2} \quad , \tag{4.30}$$

where m_i is binary fill value for the *i*-th subgrid location within the HSI pixel and *n* is the number of locations in the subgrid along a single dimension.

In the event that near real-time target detection is desired, the user may choose to circumvent the spin-image filtering process. A crude estimation of M may be calculated using the following three-step process. First, all points within the raw point cloud should be assigned a binary label according to whether or not they fall within a user-specified z-window, per Section 4.1.4.1. Secondly, all points from the raw point cloud should be projected into the HSI FPA in order to determine which points fall within a given HSI pixel. Finally, based on the points that map to a particular HSI pixel location, M may be estimated as the fraction of points that fall within the z-window. It should be noted that this technique does not provide for shape-based discrimination of points within the z-window. Furthermore, in the event that the Lidar and HSI sensors do not share the same viewing geometry, a sparse z-buffer, as described in Section 4.1.5, should be applied to the raw point cloud prior to calculating M fractions.

While the reader may recognize the pixel purity map as the desired end product of the entire target detection process, it is important to realize that the pixel purity map is based solely on 3D shape information and may therefore be riddled with false alarms. The fidelity of the map will be dependent upon scene complexity and Lidar resolution. Subsequent spectral processing may prove to be highly effective in reducing the number of false alarms.

4.3 Atmosphere Generation using MODTRAN

Developed by the US Air Force Research Laboratory, MODTRAN is a radiometric transfer code that is used to model atmospheric phenomenology. The code has numerous degrees of freedom for simulating an atmosphere, to include gaseous constituent levels, sensor-to-scene geometry, time of day, and geographic location. The output of the code includes estimates for sensor-reaching path radiances and path transmissions at a user-specified spectral range and resolution.

Eq. (1.2), restated as

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[KE'_s(\lambda)\tau_1(\lambda)\cos\theta + FE_d(\lambda) \right] \tau_2(\lambda) \left[M\frac{r_t(\lambda)}{\pi} + (1-M)\frac{r_b(\lambda)}{\pi} \right] + L_u(\lambda) \,\mathrm{d}\lambda$$

provides a fundamental framework for predicting the sensor-reaching scene radiance. $E'_s(\lambda)$, $E_d(\lambda)$, and $L_u(\lambda)$ are radiometric vector quantities. Since the irradiance terms $E'_s(\lambda)$ and $E_d(\lambda)$ are not in the same native space as the output from MODTRAN, an approximation to Eq. (1.2) is presented as

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left\{ \left[K \frac{L_s(\lambda)}{\cos\sigma} \cos\theta + FL_d(\lambda) \right] Mr_t(\lambda) + ML_u(\lambda) \right\} d\lambda + (1 - M)L_{bg}(\lambda) \quad .$$

$$(4.31)$$

MODTRAN outputs a number of files that describe the generated atmosphere. The tape7 output file may be parsed to derive the direct solar $L_s(\lambda)$, downwelled solar $L_d(\lambda)$, and upwelled solar $L_u(\lambda)$ radiance terms. $L_{bg}(\lambda)$ is an empirically-derived radiance vector that describes the local background and is further explained in Section 4.4, while the implications of the model itself are detailed in Section 4.5. Two separate MODTRAN runs are required to create a single radiance vector triplet $[L_s(\lambda), L_d(\lambda), L_u(\lambda)]$. First, a run is completed with a 0% reflector as the target material. This run serves to isolate the upwelled path contribution $L_u(\lambda)$. Second, a run is completed with a 100% Lambertian reflector and multiple scattering enabled. In doing so, the second run incorporates a direct and downwelled solar term in addition to the upwelled term. Two columns in the output tape7 file consist of the ground-leaving radiance $L_{ground}(\lambda)$ and direct solar radiance $L_s(\lambda)$. These terms may be used to estimate the downwelled path radiance via

$$L_d(\lambda) = L_{ground}(\lambda) - L_s(\lambda) \quad . \tag{4.32}$$

This two-run method produces a coarse estimate for the sensor-reaching radiance triplet. These estimates may be refined by incorporating additional scene descriptors into the MOD-TRAN tape5 card deck. Specifically, Card 1 may be modified to allow for a spectrallydependent Lambertian surface. Furthermore, the inclusion of optional cards 4A, 4L1, and 4L2 allows for separate descriptions of target and background albedo. [Berk *et al.* 2003].

When implementing an invariant target detection process, many atmospheres and radiance vector triplets need be generated as a means of capturing atmospheric variability across the scene. The viability of the invariant approach is largely dependent upon the relevancy of the predicted atmospheres relative to the in-scene atmosphere. Ientilucci provides a thorough yet succinct review of the MODTRAN input parameters that have the greatest impact on the resulting atmosphere [Ientilucci 2005].

4.4 Background Radiance Cube Generation

The physical model implemented in this research allows for the creation of signature spaces that capture the spectral characteristics of both pure and mixed pixels. In order to create a mixed radiance signature, spectral information for the local background is necessary. Based on Eq. (1.2), a local background reflectance spectrum $r_b(\lambda)$ is required as an input to the model. While an accurate target reflectance spectrum $r_t(\lambda)$ should be readily available as *a priori* knowledge, the same does not hold for $r_b(\lambda)$. Even with a coarse assessment of background classes within the image, reliable inference of class reflectance spectra may not be possible. Again, this problem is one of the primary motivators for the invariant method – that is, atmospheric compensation to transform a radiance cube into the reflectance domain may not be robust for scenes with spatially varying illumination levels.

Creating mixed signatures according to Eq. (4.31) may be advantageous as it allows for mixing of the target and background in the radiance domain. This model uses a local background mean $L_{bg}(\lambda)$ based on scene-derived radiance vectors to serve as the background contribution. A scene-derived background vector may be estimated by

$$L_{bg}(\lambda) \approx \int_{\lambda} \beta_p(\lambda) \left\{ \left[K \frac{L_s(\lambda)}{\cos \sigma} \cos \theta + F L_d(\lambda) \right] r_b(\lambda) + L_u(\lambda) \right\} d\lambda \quad .$$
(4.33)

A local background radiance cube may be generated by using the pixel purity map, described in Section 4.2.5, to mask the sensor radiance cube. Recall that the pixel purity map is a preliminary grayscale representation of target locations that is based solely on 3D shape information. The gray levels in the map correspond to the amount of the projected pixel that is filled by target points. To account for potential errors in the 3D projection of target points onto the FPA, the gray level map is converted to a binary image and then dilated. Areas in the sensor cube that have been masked are populated by taking the mean of n nearest non-masked neighbors. n should be small enough to capture only local background features, but large enough to capture local background variability. In practice, this nearest nonmasked neighbor technique for populating masked regions within the background radiance cube works equally well in both homogenous background regions and background transition regions.

4.5 Constrained Signature Space Generation

The physical model to be implemented for this research is restated as

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[K E'_s(\lambda) \tau_1(\lambda) \cos \theta + F E_d(\lambda) \right] \tau_2(\lambda) \left[M \frac{r_t(\lambda)}{\pi} + (1 - M) \frac{r_b(\lambda)}{\pi} \right] + L_u(\lambda) \, \mathrm{d}\lambda$$

The reader may recognize this model as being very similar to Eq. (3.1), with the primary difference being that the resulting radiance vector is integrated across the sensor response. Eq. (1.2) may be used to predict a single radiance signature, as measured by a spectral sensor. Localized estimates of K, θ , F, and M, in the form of feature maps as described in Section 4.2, allow for the creation of pixel-specific target signatures across the FPA. Per Section 4.3, global values for $L_s(\lambda)$, $L_d(\lambda)$, and $L_u(\lambda)$, based on a 100% Lambertian reflector for a single atmosphere, are estimated using MODTRAN. Based on atmospheric variability across a scene, the user may generate n atmospheres resulting in n radiance vector triplets. The sensor's spectral response $\beta_p(\lambda)$, solar zenith angle σ , and target reflectivity $r_t(\lambda)$ are assumed to be *a priori* knowledge. The model formulation in Eq. (4.31) allows for mixing of the target and background in radiance space. Note that the model allows for a target *reflectance spectrum* to be forward propagated into the radiance domain, while the background contribution is a scene-derived radiance estimate. This is preferable because an empirical estimate of the background radiance $L_{bg}(\lambda)$ on a per-pixel basis, as explained in Section 4.4, is likely to be more robust compared to applying an atmospheric compensation routine to transform the radiance cube into the reflectance domain. Recall that one of the initial motivations for the invariant approach was the lack of robust performance in atmospheric compensation algorithms.

In order to account for errors in the feature maps, the map values are also modulated by $\pm 10\%$, relative to the initial estimate values, generating k shadow maps, t illumination angle maps, f shape factor maps, and m target purity maps. The modulation value of $\pm 10\%$ is an arbitrary amount of variability. In practice, this amount should be based on the user's confidence level in the accuracy of a given feature map and the relative impact of the feature variable upon the integrated radiance vector. The values in the modulated feature maps are constrained between 0 - 1 for K, F, and M and $0^\circ - 90^\circ$ for θ . Finally, each pixel-specific signature space consists of $k \times t \times f \times m \times n$ radiance vectors.

Since M is an estimate of target abundance for a specific pixel, one may choose to build pixel-specific signature spaces only for pixels with M values greater than some user-defined threshold. Additionally, if one were interested in only detecting partially occluded targets, one could specify a range of M values for pixel selection.

4.6 Detection Statistic

Target detection statistics come in a variety of forms, to include pure pixel and subpixel target detection schemes. Many detection statistics are based on null and alternative hypothesis testing. For example, as specified in Section 2.3.2.2, the problem of subpixel target detection may be posed via the following hypothesis test:

$$H_0 : \mathbf{x} = \gamma B + \mathbf{n}$$
$$H_1 : \mathbf{x} = \alpha T + \gamma B + \mathbf{n}$$

The null hypothesis H_0 predicts that a pixel is comprised of a linear combination of background basis vectors B, plus a residual noise term **n** that is assumed to be mean-centered, Gaussian white noise. The alternate hypothesis H_1 models a pixel as a linear combination of both target basis vectors T and background, plus the noise term. Each column of both Band T is a basis vector from that respective subspace group [Pan and Healey 2001].

While the proposed research may be considered a subpixel target detection method, it is fundamentally different from the subpixel detectors described in Section 2.3.2.2. Previous methods generate two distinct vector spaces. A target space is derived using a pure target reflectance spectrum, atmospheric estimates, and a radiometric model. A background space is typically constructed using non-target-like scene vectors. These *global* vectors spaces are then used for processing the entire image. It is important to realize that while the background vector space may be, and probably is, representative of mixed spectra, the target vector space is inherently pure. Subpixel target detection is accomplished via application of a target detection statistic that is based on linear combinations of target and background spectra.

In contrast, the method presented here creates a unique *signature space* for each potential target pixel (as labeled by the pixel purity map), based on the *local* spatial and spectral information available at each pixel location. Depending upon the mixing level predicted for a pixel, a signature space is derived using either a pure target spectrum (*i.e.*, M = 1) or a linear combination of target and background spectra (*i.e.*, M < 1). Hence, subpixel information is incorporated into the signature space *prior* to the application of a test statistic. The advantage of incorporating subpixel information during signature space generation may prove to be significant. Target detection statistics that are based on mixed pixel hypotheses tests, such as the one above, call for the characterization of both the target and background spaces. These characterizations may be either geometric or stochastic in nature; however, accurate representation of vector spaces via either approach has proven to be problematic [Cisz 2006]. Stochastic methods often describe a space according to its mean vector and covariance matrix. Such a description only holds if the space is multivariate Gaussian. Alternatively, geometric methods model an *n*-dimensional space as a convex hull, using the end members of the hull space to describe the data. In the event that a vector space is not convex, a geometric method may fail to adequately describe the space. Furthermore, as true data spaces are typically sparse in nature, modeling the space as a single, contiguous hull may overestimate the true span of the space. Finally, given the inadequacy in current geometric methods, it stands to reason that the number of end members required to optimally describe a convex hull may vary significantly from one image to the next.

By creating a signature space, the mixed nature of a pixel is inherently built into the vector space. Furthermore, since a unique signature space is created for each pixel, the target detection statistic may be greatly simplified. Instead of comparing a test pixel to distinct

target and background spaces, the test pixel is compared to every vector in the signature space. To that end, the following detection statistic

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (x - S_i)^2}{N}}, \qquad (4.34)$$
$$\alpha_{RMSE} = \frac{1}{RMSE}$$

based on root mean squared error (RMSE), is employed. Every N-dimensional vector in a pixel's constrained signature space S compared to the pixel's actual spectrum x, with the minimum RMSE score being stored for the pixel. Recall that the proposed research will use spin-image techniques to predict which pixels are likely to contain targets. Only scene pixels with pixel purity M values estimated to be 0.33 (*i.e.*, containing at least 33% target) or greater will be considered for the α_{RMSE} test. Finally, a binary threshold will be applied to pixels with a RMSE statistic score to arrive at a target detection map.

Another option for creating a final detection map is to feed the signature space information into a traditional target detector. For this research, the Physics-based Orthogonal Subspace Projection (PB-OSP) detection statistic, formulated as

$$T^{\dagger} = (T^{t}T)^{-1}T^{t}$$

$$P_{T} = TT^{\dagger}$$

$$B^{\dagger} = (B^{t}B)^{-1}B^{t} , \qquad (4.35)$$

$$P_{B}^{\perp} = I - BB^{\dagger}$$

$$\alpha_{PB-OSP} = \frac{P_{T}P_{B}^{\perp}\mathbf{x}}{P_{T}P_{B}^{\perp}\mathbf{t}_{avg}}$$

was chosen [Ientilucci 2005]. In implementing the PB-OSP detector, a pixel-specific signature space S is substituted for T, which is traditionally a global pure target space. B is the set of global end members that describe the background. P_B^{\perp} and P_T are $N \times N$ symmetric projection operators that project a vector onto the space orthogonal to the background space and onto the target space, respectively.

The background end members B are derived by processing the scene with the MaxD algorithm [Lee 2003]. If a data set has N spectral dimensions, then the output of MaxD is N + 1 end members. MaxD is a deterministic characterization of the background space based on a progressive series of orthogonal projections in N-space. The user may select a subset of the output end members, as the output ordering of the end members is directly related to their degree of spectral distinction.

In order to prevent background contamination by the target, only scene vectors with spectral angles greater than a user-defined threshold were used as inputs to the MaxD process. The spectral angle A of a scene pixel is calculated as

$$A = \cos^{-1} \left(\frac{\mathbf{t} \cdot \mathbf{s}}{\|\mathbf{t}\| \|\mathbf{s}\|} \right) \quad , \tag{4.36}$$

where **t** is a target radiance vector and **s** is a scene radiance vector. If ground truth is available, then **t** may be supplied as a known pure target vector from the scene. Otherwise, **t** may be generated by populating Eq. (3.1) with reasonable values relative to the scene – that is, a target reflectance spectrum may be forward -modeled into the radiance domain using a scene-representative atmosphere and suggested values of 0.9, 0.0°, 0.8, and 1.0 for K, θ, F , and M, respectively. It is important to recognize that the dilated pixel purity map described in Section 4.4 should *not* be used to mask the sensor cube when trying to spectrally characterize the background. The pixel purity map is based solely on 3D shape; therefore, if multiple targets with high spectral contrast relative to one another and the background are masked prior to background characterization, then the resulting PB-OSP statistic will be high for all masked targets. This is primarily due to the fact that the P_B^{\perp} operator projects the test pixels into a space orthogonal to the background. If false alarm radiance spectra are not captured in the background characterization, then they will not be suppressed by PB-OSP.

Chapter 5

Results and Analysis

5.1 Synthetic Data

One drawback associated with the proposed multi-modal fused approach is its requirement for near-cotemporal hyperspectral and LIDAR datasets over a common scene. Furthermore, geometric parameters may only be estimated when the LIDAR spatial sampling frequency is higher than that of the hyperspectral sensor. Such data is not commonly available at present. The purpose of this research is to evaluate what value might be gained by such a data set. If the results are promising, this research may serve as justification for the multi-modal data collections, as well as the development of multi-modal sensor suites.

Due to these demanding data requirements, synthetic data sets were used in evaluating the performance of the fused detection process. RIT's DIRSIG tool provides a means for generating hyperspectral and LIDAR data sets that are physically realistic [Brown 2006]. DIRSIG's hyperspectral module is capable of creating single or multi-band images spanning the visible to long wave infrared spectrum. Passive images are created using ray-tracing techniques applied to models of the atmosphere and 3D scenes. The LIDAR module is a quasi-monochromatic extension of the hyperspectral module. Instead of using ray tracing, the Lidar utility is based on photon mapping techniques which allow for efficient estimation of time-of-flight for photons emanating from and returning to the sensor platform on a per pixel basis. An additional benefit in using DIRSIG is that scene-wide ground truth data is available.

5.1. SYNTHETIC DATA

All of the data sets presented in the subsequent results sections pertain to an actual area located at Rochester Institute of Technology, pictured in Figure 5.1. This 70 $m \times 70$ m area, commonly referred to as *Microscene 1*, consists of an open grass field surrounded on three sides by several different types of deciduous and evergreen trees. Other features in the area include an archery berm and a small shed. This area was deemed interesting from a collection and simulation standpoint due to the variety of terrain and potential for concealment in a small area.



Figure 5.1: Image of Microscene I area.

5.1.1 Ideal Lidar, Ideal Hyperspectral

The first input data set to the processes described in Chapter 4 was completely ideal. The Lidar point cloud has (x, y) postings at fixed intervals of 0.2 m, with perfect point accuracy relative to the 3D scene geometry. The point cloud is collected from a nadir viewpoint using DIRSIG's make_dem function. The point cloud is visualized in Figure 5.2. The spectral sensor



Figure 5.2: Visualization of ideal 3D point cloud false colored by height. (a) Nadir view and (b) oblique view.

is a framing array system with a response modeled after AVIRIS [Jet Propulsion Laboratory 2006]. The spectral data was also collected from a nadir geometry at a standoff range of 12.5 km. The 64 × 64 framing array is populated by 8 μm pixels with 100% fill factor. A system focal length of 100 mm yields a ground instantaneous field of view (GIFOV) of 1 m at the center of the field of view (FOV), as shown in Figure 5.3(a). Given the 0.2 m post spacing of the Lidar point cloud, the HSI pixel was oversampled by an approximate factor of 25×. A high resolution version of the cube with a GSD of 0.125 m is shown for reference in Figure 5.3(b). While the full AVIRIS response spans from 0.373 μm to 2.503 μm , data

was only captured in the visible and near infrared portions of the spectrum. This spectral cropping was intended to mimic the spectral response range of a single, silicon-based sensor. The reader should note that the values of K and F for a given scene location may change dramatically between the visible and long-wave infrared regimes. Emissive path radiances are not adequately modeled via the physical model used in this research effort. Similar to the Lidar data, the spectral radiance measurements have zero noise.



Figure 5.3: False color composite of ideal hyperspectral cube. (a) Actual HSI cube spatial resolution - 1 m GSD and (b) high resolution reference - 0.125 m GSD.

The numeric labels on the high resolution image correspond to the following ground truth:

1. Gray Humvee [†]	6. Red SUV ^{\ddagger}
2. Calibration panel	7. Gray SUV ^{†‡} hidden under tree
3. Gray SUV ^{†‡}	8. Gray SUV ^{†‡} inclined on berm
4. Gray shed	9. Gray Humvee ^{\dagger} inclined on berm

5. Red sedan

Legend: † = Target material, ‡ = 3D target shape

This scene was designed to incorporate a variety of target scenarios, ranging from targets in the open to heavily occluded targets with localized clutter. The actual target in the scene is the gray SUV at locations 3, 7, and 8. The red SUV serves as a spatial decoy and, in theory, should be eliminated from final detection maps via spectral filtering. Likewise, the gray Humvees are spectral decoys (*i.e.*, same gray paint as SUVs) and should be eliminated via spatial filtering. Another interesting scene aspect is the scale of the shrubs on the berm. In modeling these shrubs, their scales were matched to roughly that of the vehicles. In doing so, the shrubs serve as scene clutter which is unlikely to be removed during the zwindowing process, thus presenting a challenge to subsequent spin-image filtering. However, the greatest challenge to both the spatial and spectral detection processes is target 7. This SUV is located in a heavy shadow and is 50% occluded relative to a sensor at nadir.

5.1.1.1 Point Cloud Processing Results

Each of the processes described in Section 4.1 produce a set of 3D points with associated feature labels. The sparse shadow feeler returns a subset of the original point cloud consisting of all points which are believed to be in shadow. The ground point extraction and normal estimation process returns an interpolated set of points that are believed to be representative of the ground plane and the ground points' associated normal vectors. The sparse skydome feeler returns the entire original point cloud with an *F*-value assigned to each point. Finally, the target point estimation process returns a subset of the original point cloud consisting of potential target point samples.

Section 5.1.1.2 provides a sufficient visualization of these process outputs; however, the output of the target point estimation process warrants additional attention. In the event that the Lidar-based target point estimation process *misses* a target point; then the proposed method is likely to fail in detecting the point in subsequent spectral processing. Put differently, the Lidar data serves as a preliminary cue for subsequent spectral target detection. For this reason, the Lidar-based target point detection process can tolerate a high false alarm rate, placing additional burden on the spectral detection process.

Recall that per Section 4.1.4.3, the output of the spin-image process is a list of correspondence points, from which scene target points are extracted. Visualization of the correspondence point locations relative to the extracted target point locations is imperative to ensure that the target point detection algorithm is working properly. Noting correspondence locations is also likely to give an analyst a better idea of scene features with the 3D shape of the target, as the correspondence point density is likely to be higher in true target areas versus false alarm areas.

Figures 5.4-5.5 show the outputs of the target point extraction process, as described in Section 4.1.4. Note the positions and scales of the objects in the figure are relative to an arbitrary global coordinate system. These plots are projections of the actual points in 3D space, prior to creation of a pixel purity map. Figure 5.4 shows the scene points that remain after z-windowing above the ground plane. After normalizing the ground plane, z-



Figure 5.4: Target point extraction results: z-windowed ideal point cloud.

thresholds of 1.0 m and 3.5 m were used as lower and upper thresholds, respectively. Figure 5.5 illustrates the output of the spin-image filtering process. All of the z-windowed points are labeled as either a scene-correspondence (yellow), extracted target point (red), or non-target point (black). Note that while many of the targets in the scene have high correspondence point densities, other man-made features within the scene also exhibit similar correspondence densities. Given the low correspondence density on target 7 (*i.e.*, SUV partially hidden by tree) compared to targets 8 and 9 (SUV and Humvee surrounded by tall shrubs), the results indicate that the performance of spin-image filtering is more resistant to clutter in close proximity to a target than to target occlusion; however, such performance is largely



Figure 5.5: Target point extraction results: scene correspondences (yellow) and extracted target points (red) for ideal point cloud.

dependent upon the cloud point density, point accuracy, and the spin-image generation criteria.

The spin-image generation criteria used to generate the yellow correspondence points are provided below.

- Spin support = $4.0 \ m$
- Bin size = 0.2 m
- Spin angle = 20°

One interesting aspect in computing Figure 5.5 is the fact that the spin angle criterion was applied to *both* the model and scene. Per Section 4.1.4.2, this criterion is typically only applied when creating the model spin-image library, for the purpose of adding self-occlusion effects to the library. Based on the assumption that most of the Lidar samples would fall on the hood and roof of the SUV, the spin angle criterion was also applied to the scene spin-images, allowing for a fair amount of clutter reduction without adversely affecting the detection of actual target objects.

5.1.1.2 Map Creation Results

The following maps were created via the processes described in Section 4.2. Figure 5.6 illustrates the map output for the ideal Lidar data. Again, the maps in (a), (b), (c), and (d) represent local estimates of the terms K, θ , F, and M from Equation (1.2). The sun's azimuthal position for this DIRSIG scene was 207°, as measured clockwise from the +y axis. For the remainder of this paper, the y axis will refer to the axis running along the vertical dimension of the page, and the x axis along the horizontal. The solar zenith (*i.e.*, declination), as measured clockwise from the +z axis, was approximately 33°. With this information in mind, the shadow map output in Figure 5.6(a) appears to be reasonable. Shadow features fall away from the solar direction and are most prominent for tall scene objects, such as trees. Furthermore, the heavy shadowed regions "line up" with those in the false color composite, per Figure 5.3. The gray regions indicate some partial shadowing effects across the berm area, as well as off the edges of some of the vehicles.

The incident illumination angle map in Figure 5.6(b) was derived by sampling the actual ground geometry used in creating the DIRSIG scene. That is, all object geometry instances

5.1. SYNTHETIC DATA



Figure 5.6: Feature maps for ideal Lidar data. (a) Shadow map (b) incident illumination map (c) shape factor map and (d) pixel purity map.

were removed from the gdb file to create a simple terrain model. Then, make_dem was applied to the terrain to create a terrain point cloud with 0.2 m post spacings. As expected, the map resembles a traditional terrain height map, with flat regions having a constant gray level. The dark areas on the berm indicate that the berm normals are pointing in the general direction of the sun, allowing the reader to ascertain the azimuthal position of the sun by inspection.

Figure 5.6(c) shows the output map for the shape factor term. The sparse skydome feeler sampled the skydome in azimuth and zenith increments of 30° and 15° degrees, respectively, for a total of 72 directional samples. The gray levels in the map correspond to the amount of the skydome that is visible at any particular location in the scene. As such, one might expect the brightest values to occur in the center of open areas, as well as on the tops of trees. This appears to be the case, while the darkest areas occur in ground regions that are immediately surrounded by foliage.

Finally, Figure 5.6(d) provides the output for the pixel purity map. The gray levels were achieved by mapping all of the scene's correspondence and extracted target points onto the HSI FPA, per Section 4.2.5. The reader should notice immediately that all of the man-made features in the scene are prominent, with the exception of the calibration panel which was eliminated by the z-windowing process. This is most likely due to the fact that many of these features contain one or more large, planar surfaces, resulting in a probable spin-image match. Note that the correspondence and extracted (*i.e.*, yellow and red) point densities in Figure 5.5 translate to the pixel purity levels in Figure 5.6(d).

5.1.1.3 Signature Space Generation Results

After creating spatial feature maps, several additional inputs are required in order to produce a unique spectral signature space for every pixel location in the HSI FPA. In building the hyperspectral image cube using DIRSIG, the exact set of atmospheric parameters for the scene was known. The atmospheric parameters used to generate the scene are the same as those used to create the path radiance vectors described in Chapter 3. As such, only one path radiance triplet was used to create the signature spaces – that is, all variability in the target spaces is due to modulation of the geometric terms in Eq (1.2).

In implementing physics-based approaches on real data sets, atmospheric parameters such as visibility, amount of water column vapor, etc. are often unknown. This results in modeling many different atmospheres using MODTRAN and ultimately increases the number of the vectors comprising the target space. The number of MODTRAN runs required to adequately characterize the atmospheric variability across a scene is largely scene dependent. Ientilucci provides a thorough description of characterizing atmospheric variability for the Forest Radiance data set in [Ientilucci 2005].

In order to account for the fact that the values at any given location in the feature maps are likely to have some error, the maps themselves are modulated by varying degrees. Based on the variable sensitivity study described in Chapter 3, it was most prudent to include the most variability on the feature maps that are likely to have the greatest impact on the resulting signature space, namely the shadow and pixel purity maps. Specifically, five maps were used for both the shadowing K and pixel purity M terms. Using the preliminary estimate maps as a baseline, modulation factors of ± 0.2 and ± 0.1 were added, while all final map values were constrained between 0 and 1. Three maps were used for the incident illumination angle term θ . Given the solar zenith angle of 33°, the preliminary map was modulated by ±3.5° in order to vary the solar loading term by approximately 10%. All values for θ were constrained between 0° and 90°. Finally, because the variable sensitivity study indicated the shape factor term had little to no impact on an integrated path radiance spectrum, no modulation of the shape factor map was deemed necessary.

A background radiance cube is empirically derived by masking the raw radiance cube with the pixel purity map and subsequently filling masked locations with the mean vector for n nearest non-masked neighbor pixels, as described in Section 4.4. Figure 5.7(a) clearly



(a)

(b)

Figure 5.7: Background map creation. (a) Sensor cube masked by pixel purity map and (b) background radiance cube false color composite.

illustrates the effect of using the pixel purity map in Figure 5.6(d) to mask out potential

target locations. Again, the mask is created by converting the pixel purity map to a binary image and then dilating it to mitigate projection registration errors. The result of filling the masked locations with the mean of the three nearest non-masked neighbors is shown in Figure 5.7(b). The resulting fill vectors for the local background appear to be reasonable across the image, to include background transition areas, such as the transition from grass to sand. Due to the fact that the calibration panel was *not* included in the pixel purity map and the relative proximity of target 5 to the panel, some spectral contamination of the background has occurred at the position of target 5.

A lower pixel purity limit of 30% target was used in creating signature spaces. For pixels with M > 0.3, a signature space consisting of 75 vectors was created. Again, the total number of vectors is based on the fact that 1 atmosphere, 5 shadow maps, 3 illumination angle maps, 1 shape factor map, and 5 pixel purity maps were used to capture scene wide variability, as is illustrated in Figure 5.8. Note that 1 atmosphere was used to test the viability of spatially-constrained approach under a known atmosphere. In practice, a user should include more than one atmosphere, as perfect knowledge of atmospheric conditions across an image is improbable. Each pixel's signature space is unique, derived from the spatial characterization of the scene via Lidar processing. Figure 5.9 demonstrates how the spaces can vary based on such spatial characterization. As should be expected, the overall magnitude of the vector space in Figure 5.9(a) for a fully exposed SUV (*i.e.*, target 3) is significantly higher than that for a heavily shadowed SUV (target 7) in Figure 5.9(b). Furthermore, the general shapes of the spaces appear to be different due to the difference in integrated path radiance contributions. These two spaces also reinforce the fact that the addition of terms to a physical model can lead to unnecessary space variability for a given



Figure 5.8: Block diagram of signature space generation process for microscene.

pixel if constraints are not applied. Without the application of Lidar-derived constraints, a global target space would be represented by the union of all of the pixels' signature spaces, to include those in Figure 5.9(a) and (b).

Figure 5.10 shows the reduction in the span of a signature space that results from constraining the physical model. The red curves represent the constrained signature space for a fully-resolved pixel on target 1 (gray Humvee in the open), while the blue curves show the unconstrained signature space for the same pixel – that is, the geometric terms in the physical model were allowed to vary through the entire possible range of values in creating

5.1. SYNTHETIC DATA



Figure 5.9: Example signature spaces. (a) Space for pure pixel on target 3 and (b) space for pure pixel on target 7.

the unconstrained space.



Figure 5.10: Constrained (red) and unconstrained (blue) spectral signature spaces.

5.1.1.4 Detection Results

Inverse RMSE

One of the simplest forms of a target detector is based on root mean square error. In the event that the various terms feeding into the physical model are correct, then at least one of the resulting target vectors should closely match the sensor's measured spectrum for the particular pixel. Implicit in this assumption is that the synthesized target space is in the same calibrated space as the sensor. An inverse RMSE score is calculated by comparing every signature vector in a pixel's target space to the corresponding pixel's spectrum via Eq (4.34). The signature vector with the highest inverse RMSE score is then recorded as the best match from the synthesized space. Figure 5.11 shows the best match spectra for several different locations in the image. The best match signature is represented by a dashed red curve, while the actual pixel spectrum is a solid black curve.

The reader should immediately recognize that the first two matches line up with their respective pixels' curves fairly well; however, a common discrepancy occurs in the spectral range between 400-500 nm, which is believed to be caused by the manner in which the raw path radiances are estimated using MODTRAN. A preliminary study was accomplished showing that a downwelled path radiance vector produced by a single MODTRAN run, regardless of the scattering algorithm configuration and/or target/background reflectance characteristics within the tape5 file, produces an underestimate of the downwelled path radiance when compared to DIRSIG in the 400-500 nm range for this particular scene. An overview of this study is presented in Section 7.1 of the Appendix. As such, the two-run method for estimating a radiance vector triplet, as described in Section 4.3, was implemented



Figure 5.11: Illustration of the best match spectra from several different signature spaces. (a) Target 3 (b) target 7 and (c) target 6.

for generating target spaces, recognizing that discrepancies may occur in the blue/green portions of the resulting target spectra.

Figure 5.11(c) is clearly a poor match. This is due to the fact that the target space

was generated based on a gray target reflectance spectrum. Recall that Target 6 is a red SUV. As such, this example demonstrates how the inverse RMSE metric can suppress targets that do not have both the correct shape and spectral characteristics. Finally, all scores are normalized by the maximum inverse RMSE score for the entire image to create the following detection map in Figure 5.12(a). Figure 5.12(b) provides a reference for the actual ground



Figure 5.12: (a) Inverse RMSE score map for ideal data set and (b) Truth map for resolved (green) and subpixel (red) targets.

truth. Recall that signature spaces were only generated for pixel locations with M > 0.3, per the pixel purity map output. This allows for many pixels in the hyperspectral cube to be eliminated from consideration. The spectral detector's reliance on the pixel purity map has both advantages and disadvantages. While many clutter and other non-target pixels may be thrown out, the potential also exists for true target pixels to be thrown out if they are not detected by the spin-image detection process.

The detection map score map in Figure 5.12(a) indicates that the signature spaces generated for all of the target locations had at least one vector that was a quality match to the actual pixels' spectra, to include mixed pixels on the edges of the targets. However, some of the edge pixels on the shed at location 4 and several clutter locations on the archery berm also had high scores. These high scores are most likely due to a low target mixing level (*i.e.*, $M \approx 0.3$) compared to the background.

Ground truth for the synthetic scene was generated using a standard DIRSIG output called the first material map [Brown 2006], shown in Figure 5.13(a). Each color in the map corresponds to a unique material within the scene. Spectral variability across homogenous material regions is achieved through the application of a texture map. This particular material map product is created by tracing a ray from each detector element through the sensor optical system and intersecting the ray with a scene object. Then the material map pixel corresponding to the detector element is filled with the material index associated with the intersected object. Since the material index assigned to the SUV's gray paint is known, the material map may be processed to show only locations where the gray paint is present. In order to determine subpixel truth locations, the scene was spatially oversampled for the creation of the material map. Then, target purity fractions were calculated using the same process described in Section 4.2.5. Subpixel targets were defined as those pixels with less than 70% fill by the target object (*i.e.*, gray SUV), producing the pixel purity truth map shown in Figure 5.13(b). Non-resolved and resolved target locations are colored red and green, respectively. Note that the gray Humvee locations have been manually eliminated from the pixel purity truth map.



Figure 5.13: Ground truth maps. (a) Oversampled first material map and (b) Pixel purity truth map.

PB-OSP

While the inverse RMSE detection metric appears to perform relatively well on the ideal data set, the Physics-based OSP (PB-OSP) statistic, specified in Eq. (4.35), is suggested as a more robust detection metric. The most significant aspect of orthogonal subspace detectors is their projection of the test pixel onto a space that is orthogonal to the background. In doing so, these detectors excel in suppressing all pixels that resemble background and should therefore be ideal for eliminating the clutter locations that scored well for the inverse RMSE detector.

Instead of implementing Ientilucci's method of supplying a global target space for T, the current research uses each pixel's custom signature space. The background was characterized geometrically through the use of 5 end members. A false color composite of the selected end members is shown in Figure 5.14 below. Many algorithms such as MaxD, SVD, or N-FINDR
5.1. SYNTHETIC DATA



Figure 5.14: False color composite of background end members. (a) Sand (b) tree (c) red paint (d) grass and (e) shadow.

exist for automatically determining the end members for a scene [Cisz 2006]. Typically these methods mask target-like pixels from the scene prior to end member extraction. For the sake of controlling the detection process, the end members in Figure 5.14 were selected by hand via inspection of the image. The resulting PB-OSP detection map is provided in Figure 5.15. Compared to the inverse RMSE score map in Figure 5.12, the PB-OSP statistic performs



Figure 5.15: PB-OSP score map for ideal data set and (b) Truth map for resolved (green) and subpixel (red) targets.

significantly better across the image as a whole. The previous map's false alarm locations are suppressed by the orthogonal background projection operator. One pixel location on the shadowed SUV in target location 7 has been suppressed due to the inclusion of a shadow end member; however, it is interesting to note that the relative detection score for the other pixels on target 7 are on the same scale as pixels on the less challenging targets. This indicates that the detection score map could be thresholded such that all background and clutter pixels would be eliminated from a final detection map.

Finally, due to the small number of pixels on true targets, traditional Receiver Operator Characteristic (ROC) curves have not been produced. The detection statistic score maps are a preferable alternative to ROC curves because they allow for visualization of detector performance at each pixel location, potentially providing greater insight into the strengths and weakness of a given detection statistic.

Comparison with Ientilucci's Method

As described in Section 2.3.2.2, Ientilucci's development of

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[\left(\frac{E_s(\lambda)}{\cos \sigma'} \cos \sigma_{new} \tau_1(\lambda) + F E_d(\lambda) \right) \tau_2(\lambda) \frac{r(\lambda)}{\pi} + L_u(\lambda) \right] d\lambda$$
(5.1)

represents one of the most current physical models being used for application of the invariant method [Ientilucci 2005]. As such, Ientilucci's model and methodology in general provides a benchmark to which the proposed research is compared. Relative to the Ientilucci's model, the proposed model

$$L_p(\lambda) = \int_{\lambda} \beta_p(\lambda) \left[KE'_s(\lambda)\tau_1(\lambda)\cos\theta + FE_d(\lambda) \right] \tau_2(\lambda) \left[M\frac{r_t(\lambda)}{\pi} + (1-M)\frac{r_b(\lambda)}{\pi} \right] + L_u(\lambda) \,\mathrm{d}\lambda$$

has additional complexity in the inclusion of a shadowing term K and mixing term M. With the additional information provided by cotemporal Lidar data, the proposed model may be constrained to produce signature spaces which may be more representative of target pixels. Specifically, one might expect the proposed model to outperform Ientilucci's on heavily shadowed targets. While Ientilucci is able to vary the solar loading term using the σ_{new} term, his formulation couples the solar loading variability to $\pm 10\%$ of the direct solar term [Ientilucci 2005].

For point of comparison, a global target space was generated for the microscene hyperspectral cube, consisting of three vectors. The correct atmosphere was used in creating the target space, while a fixed value of 0.8 was used for F. The variability in the space comes from modulation of the direct solar term by $\pm 3.5^{\circ}$ in order to modulate the solar loading term by approximately 10%. The same end members were used to describe the background. Figure 5.16(a) illustrates the result of running the PB-OSP statistic on the same pixel locations that were used in creating the detection map in Figure 5.15. Recognizing that Ientilucci's method would not normally benefit from a preliminary spatial cue, Figure 5.16(b) shows the detection results when PB-OSP is applied to the entire hyperspectral cube using the global target and background spaces.

Several significant results are evident in these detection maps. First, the reader should recognize the benefit provided by the proposed preliminary spatial cue. Not only are many



Figure 5.16: PB-OSP score maps for Ientilucci's global target space method. (a) Spatiallycued detection scores, (b) non-cued detection scores, and (c) Truth map for resolved (green) and subpixel (red) targets.

background pixels eliminated from consideration, but also some false alarm locations may be thrown out as well. Such is the case for calibration panel at location 2. The z-windowing process throws out the Lidar samples from the calibration panel, as they do not fall into a specified height window relative to the ground plane. Ultimately, the Lidar samples from the panel are not considered in creating the pixel purity map and subsequent spectral detection map. If the calibration panel is considered by the spectral detection algorithm, it scores highly due to its spectral distinction relative to the background end members. Secondly, the shadowed SUV at target location 7 does not result in a high detection score. This is primarily due to the fact that Ientilucci's implementation does not have sufficient variability so as to incorporate heavily-shadowed targets. It is not immediately clear as to whether the inclusion of an unconstrained shadowing term in Ientilucci's model would be beneficial. Without the means to constrain such a shadowing term, the added variability in the target space may decrease the distance between the target and background spaces, ultimately reducing the detector performance.

Tables 5.1-5.3 provide a quantitative comparison of the target detection results using the constrained, fused method and Ientilucci's approach, based on the PB-OSP detection statistic. Detection thresholds were selected by inspection of the PB-OSP score maps in Figures 5.15 and 5.16(b). Admittedly, all results based on these thresholds are somewhat arbitrary; however, assuming that reasonable threshold values are selected, then the performance comparison should still provide insight as to the strengths and weaknesses of the respective detection methods.

Two different threshold levels were applied for each target detection algorithm in order to assess detection performance for both non-resolved (*i.e.*, mixed) and fully-resolved (*i.e.*, pure) target pixels. For the fused method, mixed and pure detection thresholds were set at 0.17 and 0.21, respectively. For Ientilucci's method, these thresholds were set to 0.20 and 0.25. The fused thresholds were purposefully set lower than those used for Ientilucci's method in order to allow for decreased detection scores due to mixing within the target space. Tables 5.1 and 5.2 present a pixel-based detection analysis, while Table 5.3 is an assessment of object-based detection.

Table 5.1 shows the pixel-level target detection results for both detection methodologies at both threshold levels, as well as the corresponding truth information. The two detection processes perform equally well for the targets that are fully-illuminated. However, target 7 is not detected without the benefit of fused spatial information. As should be expected, many subpixel targets are undetected by either method. These missed detections are likely due to extremely low target purity levels. Table 5.2 illustrates the reduction in false alarms

			DETECTIONS		DETECTIONS	
		111	FUSED		HSI ONLY	
Target #	Mixed	Pure	Mixed	Pure	Mixed	Pure
3	9	4	6	4	5	4
7	8	2	1	1	0	0
8	11	4	2	3	3	3
Totals	28	10	9	8	8	7

Table 5.1: Pixel-level PB-OSP performance comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data.

that results from the fusion of spatial information. Note that approximately 99% of the false alarm pixels for the fused method come from the spectral decoy targets 1 and 9 (*i.e.*, the gray Humvees). Since the spin-image spatial filtering process was unable to eliminate these targets (due to the mean post spacing of the Lidar sensor), they scored highly during the secondary spectral filtering.

METHOD	THRESHOLD LEVEL	# OF FALSE ALARMS
Fused	Mixed	22
Fused	Pure	13
HSI only	Mixed	39
HSI only	Pure	24

Table 5.2: Pixel-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data.

Finally, Table 5.3 presents the detection comparison differently, in terms of whether or not at least one pixel on a target object was detected (*i.e.*, yes/no). Since both the fused approach and Ientilucci's method claim to be subpixel detectors, the subpixel detection thresholds described above were used to generate this table. All manmade objects from the scene are included. Correct detection answers are colored green, while incorrect answers are red. Again, the reader should recognize that Ientilucci's method fails on the calibration panel and the SUV hidden under the tree.

Note that the Humvee spectral decoys at locations 1 and 9 are detected by both approaches. Since these decoys have the same gray paint as the true target (*i.e.*, gray SUV), the fused detection method relies on the fidelity of the spin-image filtering process to eliminate the Humvees from the preliminary spatial cue. Again, with a higher point cloud post-spacing, distinguishing between the 3D shapes of the Humvee and SUV via spin-image processing may be achievable [Johnson 1997]. Conversely, since Ientilucci's approach is based solely on passive information, it has no means of spatially filtering decoys.

	FALSE	DETECTED	DETECTED
	ALARM	FUSED	HSI ONLY
Target 1	Y	Y	Y
Target 2	Y	N	Y
Target 3	N	Y	Y
Target 4	Y	N	N
Target 5	Y	N	N
Target 6	Y	N	N
Target 7	N	Y	N
Target 8	N	Y	Y
Target 9	Y	Y	Y

Table 5.3: Object-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data (green = correct, red = incorrect).

Additional analysis based on the ideal Lidar/HSI data set is included in the Appendix. Section 7.2 describes how the target space generation process changes with varying atmospheric conditions. Specifically, as visibility levels across a scene change, the relative importance of the spatial feature maps used to constrain the physical model may also shift. Additionally, further comparisons between the fused method and Ientilucci's approach are presented.

Computational Run Time

The ideal point cloud consisted of 122,500 points. Some of the point cloud processes were computationally trivial, while others were quite burdensome. Before documenting computer run times, it is important to point out that all of the IDL-based codes were written solely for functionality. Very little effort was put into optimizing the codes' efficiencies. That said, the run times associated with the point cloud processing codes are summarized in Table 5.4. The PC in question was a Pentium IV machine with a 3.0 *GHz* processor and 3 *Gb* of RAM.

Process	PC	Condor
Sparse shadow feeler	21 hrs	2 hours
Ground point extraction	$5 \min$	-
Ground normal estimation	$15 \min$	-
Sparse skydome feeler	?	2 days
z-windowing	43 min	-
Spin-image model library creation	1 min	-
Spin-image target point extraction	19 min	-

Table 5.4: Point cloud processing run times (approximate).

Condor is a networked computer cluster, capable of distributing a single task across multiple machines and reassembling the data upon completion. Utilization of the Condor high throughput computing system at RIT rendered the computation and analyses for this trade study tractable [Epema *et al.* 1996]. Finally, the run times associated with the creation of 2D feature maps, MODTRAN atmospheres, and signature space synthesis were all minimal. The clear outlier in Table 5.4 is the sparse skydome feeler with a run time of roughly 2 days on the networked computer cluster. This particular implementation of the sparse skydome feeler sampled the skydome above every point 72 times. The run time associated with this algorithm could be reduced significantly by reducing the number of skydome samples per point, albeit at the expense of shape factor estimate accuracy. Note that a variable sensitivity analysis, as is presented in Chapter 3, provides a means to assess the level of accuracy needed for any given geometric term in the radiometric model. For the case of the 23 km visibility atmosphere that is used for the data sets presented thus far, a reduction in the number of skydome samples per point should not have a significant impact on the detection scores. Additional suggestions for improving the speed of the methods based on sparse ray tracing are provided in Chapter 6.

5.1.2 Ideal Lidar, Noisy Hyperspectral

While the results for ideal data described in Section 5.1.1 are promising, additional experimentation was required in order to assess how the process degrades in the presence of noise. The first of such experiments involved adding spectrally correlated noise to the hyperspectral cube and fusing it with the ideal Lidar point cloud. In adding noise to the ideal HSI cube, dark noise statistics from the Modular Imaging Spectrometer Instrument (MISI) were mimicked using a technique based on principle components analysis [Peterson 2004]. The MISI sensor was chosen because system dark data were readily available and due to the fact that appreciable system noise levels have been documented [Feng 1995].

Figure 5.17 illustrates the process of generating spectrally correlated noise. If spectrally correlated dark data are available, then a principle components (PC) transform may be

implemented to transform the data into a minimally correlated space. Once in PC space, the eigenvalues for the covariance matrix represent the variance of the data in each band. The square root of the eigenvalues may be used to specify the width of Gaussian distributions from which random noise vectors are created. Next, the inverse PC transform is applied to bring the newly generated noise vectors back into the original spectrally correlated space. Finally, the noise vectors may be scaled to the user's liking and interpolated to the same band centers as the hyperspectral image cube. In the event that interpolation is required, care should be taken to ensure that the spectral range of the input dark data is wider than that of output cube; otherwise, interpolation artifacts may occur at the spectral extremes.



Figure 5.17: Block diagram for generating spectrally correlated noise.

Figure 5.18 shows the correlation matrices for the input MISI dark data and the output synthesized noise. Note that 15,000 random noise vectors were created in order to build

up the statistics for the output noise. Then, for a hyperspectral cube consisting of n pixel elements, n noise vectors were randomly drawn from the set of 15,000. Figure 5.19 shows



Figure 5.18: Spectrally correlated noise: correlation matrices. (a) Input dark data and (b) output noise vectors.

the result of adding spectrally correlated noise to the ideal hyperspectral cube. The signal



Figure 5.19: False color composite of hyperspectral cubes. (a) Original ideal cube and (b) ideal cube with spectrally correlated noise added.

to noise ratio (SNR) for the resulting HSI cube ranged from 156 at 579 nm to 4 at 943 nm. While an SNR of 4 may seem extremely low, it is important to recognize that 943 nm is a water absorption line, resulting in very little received signal after atmospheric filtering.

Detection Results

Figure 5.20 shows the effect of adding spectrally correlated noise to the hyperpsectral data. The output of the inverse RMSE map is consistent with its noise-free counterpart in



Figure 5.20: Detection score maps for ideal Lidar and noisy HSI cube. (a) Inverse RMSE, (b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.

Figure 5.12. However, the effect of adding spectral noise is clearly seen across the PB-OSP score map in the degradation of score consistency across any given target. In particular, many of the highest scores occur on mixed, edge pixels, despite the fact that mixing the target spectra with the local background should reduce the resultant score. Again, if the local background has been adequately characterized by the set of background end members, then the orthogonal projection operator should suppress those pixels with significant background

mixture levels. That said, for the case of spectrally correlated HSI data fused with a low noise Lidar point cloud, the inverse RMSE map appears to be a better detection statistic, as the clutter suppression provided by the PB-OSP statistic may be minimal. In order to make a more definitive assessment, additional applications of the presented fused technique across a variety of scenes is necessary.

5.1.3 Noisy Lidar, Ideal Hyperspectral

The second experiment in assessing the fused approach's sensitivity to noise involved adding noise to the Lidar point cloud and fusing it with ideal HSI data. In generating the noisy Lidar point cloud, DIRSIG's Lidar simulation utility was used. DIRSIG is capable of modeling various error sources including platform position uncertainty, sensor pointing uncertainty, and range quantization. The simulated Lidar system consisted of a scanned frame array. The GIFOV for a single array pixel was 0.2 m. For each scan location, a Gaussian spatial profile beam illuminated the entire frame, with the 1σ full width of the Gaussian profile being set to the size of the frame's IFOV as projected to the ground. A pulse's temporal profile was also Gaussian with a 1σ full width of 1.0 ns. The level of range quantization on the system clock was 0.5 ns or approximately 7.5 cm. Zero platform pointing/position error was added to the cloud. The amount of frame-to-frame overlap between pulses/scan positions was set to 1/4, resulting in higher density post spacings in overlapped regions. The scan rate of the system was set in order to match the along- and across-track post spacings as closely as possible. Finally, the full scan swath of the Lidar system was much greater than the spatial extent of microscene I. In doing so, the effect of along-track post spacing compression near the end of scan lines is minimized.

5.1.3.1 Point Cloud Processing Results

Similar to the presentation of the results for the ideal data in Section 5.1.1, it is prudent to begin with the actual output of the target point detection process. Prior to running the z-windowing and spin-image based target point extraction processes, the raw system point cloud was downsampled in an attempt to homogenize the post spacing density across the cloud. The downsampling process consisted of (1) specifying a desired grid with constant postings of 0.2 m and (2) using an (x, y) nearest neighbor search technique across the raw point cloud to retain the closest point to each grid location. Note that the resulting points were *not* interpolated in order to maintain the 3D location accuracies of the output points.

From the initial stages of applying a z-window to the downsampled Lidar point cloud, the effects of range quantization were observed. In particular, the overall width of the zwindow had to be increased in order to preserve all of the samples from the target vehicles. Lower and upper z-thresholds of 0.5 and 3.75 m, respectively, were applied to the data. In opening the z-window, the number of clutter points increased significantly, particularly so from the shrubs along the berm. Figure 5.21 is a nadir oriented 2D visualization of the 3D point cloud after applying the z-window. In comparing the noisy z-windowed point cloud to its ideal counterpart in Figure 5.4, the reader should recognize that the point density in the z-windowed cloud along the berm area is much higher after opening the z-window.

The negative impact of range quantization upon the spin-image detection process was initially surprising. While the mean post spacing of the noisy Lidar cloud was roughly equivalent to that of the ideal point cloud, the target point detection performance was significantly reduced, as is shown in Figure 5.22 below. Notice that the correspondence densities across the targets in the open are still quite high, while the more challenging targets



Figure 5.21: Target point extraction results: z-windowed noisy point cloud.

have far fewer correspondence points than those in Figure 5.5. The spin-image generation criteria used to generate the correspondence and extracted target points in Figures 5.22 and 5.23 are as follows:

- Spin support = 4.0 m
- Bin size = 0.25 m
- Spin angle = 45°

Many different combinations of spin-image generation criteria were used in processing the



Figure 5.22: Target point extraction results: scene correspondences (yellow) and extracted target points (red) for noisy point cloud.

z-windowed point cloud; however, all the runs produced similar results with minimal discriminability between the target and clutter point samples. Relative to processing the ideal point cloud, the bin size and spin angle criteria were increased to account for increased uncertainty in point location accuracy.

One striking aspect in Figure 5.22 is the fact that virtually all of the samples along the berm have been labeled as target samples. For the ideal point cloud, many of the clutter points along the berm were eliminated by applying the spin angle criterion when forming the scene-based spin-images. For the case of the noisy Lidar data, application of the spin angle criterion has little to no effect. This is largely due to range quantization effects, which can significantly affect the calculation of point normals and the resulting scene spin-images. A qualitative study of the impact of range quantization on spin-images is included in the Appendix. Admittedly, range quantization is not the only common source of sample location error for Lidar systems. Other system specific sources include pointing error, trigger error in the range clock, and a the lack of exact knowledge of the sensor platform location. All of these error sources add in a cumulative sense to the location error for a given point.



Figure 5.23: Target point extraction results: scene correspondences (yellow) and extracted target points (red) for ideal point cloud with wide z-window.

To be fair, the lack of false alarm points in the correspondence and target points illustration in Figure 5.5 is largely due to the fact that a narrow z-window was used. Figures 5.22 and 5.23 show how opening the width of the z-window can degrade the performance of the spin-image detection process, regardless of noise levels in the data set.

5.1.3.2 Map Creation Results

Figure 5.24 shows the 2D feature maps that were derived from the noisy Lidar point cloud and ultimately used in the signature space generation process. The scan swath of the simulated Lidar sensor was not wide enough to cover the entire FOV of the HSI sensor; hence, missing data regions are apparent across the top and right sides of the maps. Also note that the ground plane from the ideal data set was used for this experiment. This was primarily due to the fact that the simulated noisy Lidar point cloud had a significant number of multiple-bounce photons, resulting in many false returns located below the actual ground plane. It is important to recognize that the noisy point cloud had not been processed with any preliminary heuristics for eliminating false returns. These heuristics typically consist of calculating local statistics in the range dimension of the point cloud to eliminate statistical outliers below a user specified threshold such as a local median. These preprocessing steps result in degradation of a cloud's post spacing. As such, use of the ideal ground plane became an attractive option. While the accuracy of the illumination angle map does improve from using the pristine data, the impact of adding noise to the ground plane points is not overly significant due to the averaging of point normal vectors across a given HSI pixel. The robustness of the illumination angle map is demonstrated with real data in Section 5.2.

While the impact of adding spatial noise to the point cloud is evident across the maps, it is most significant in the pixel purity map in Figure 5.24(d). This figure indicates that the spin-image filtering process may not be as effective in the presence of realistic range accuracy error. The high false return density across the archery berm results in erroneously high pixel purity levels. Again, the conceptual strategy in target point detection and ultimately, producing the pixel purity map was to tolerate false alarms for the sake of not missing any true target points. Clearly, this pixel purity map reflects the increased burden being placed on the spectral target detection process.

Fortunately, the shadow and shape factor maps in Figure 5.24(a) and (c) indicate that the addition of spatial noise and range quantization had little impact on the fidelity of the resulting feature maps. Compared to the ideal shadow and shape factor maps in Figure 5.6, the gray levels are roughly equivalent. Furthermore the incorporation of variability about the map values during the signature space generation process should produce similar target spaces. The effect of preprocessing and downsampling the noisy point cloud is apparent in the presence of dropout pixels in Figure 5.24(c). Preprocessing consisted of applying a simple z-threshold to the original point cloud prior to ground plane normalization. This step eliminated many of the false returns located below flat regions of the terrain; however, the samples from the lowest section of the true terrain were also removed, resulting in the large dropout region along the upper edge of the map. Additional dropout points resulted from downsampling the noisy point cloud via the nearest neighbor method described earlier. If the nearest neighbor to a grid location happened to be a false return from below the ground plane, then the amount of the skydome visible to the point would be close to zero. In practice, a user could eliminate false ground points below the ground plane by applying a local median filter to the point cloud, flagging points with z-values greater than 2 standard deviations below the local median.



Figure 5.24: Feature maps for noisy Lidar data. (a) Shadow map, (b) incident illumination map, (c) shape factor map, and (d) pixel purity map.

5.1.3.3 Detection Results

Figure 5.25 shows how the addition of spatial noise impacts the resulting target detection score maps. Again, since spectral signature spaces are generated based on the pixel purity



Figure 5.25: Detection score maps for noisy Lidar and ideal HSI cube. (a) Inverse RMSE, (b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.

map in Figure 5.24(d), the reader should immediately recognize that the task of background suppression has shifted primarily to the spectral detection statistic. Clearly, PB-OSP performs more reliably for this scenario. The inverse RMSE statistic appears to ring on many edge pixels where significant mixing with the local background occurs, regardless of whether or not the pixel in question is actually a target. To minimize this effect the minimum pixel purity level considered for signature space generation and subsequent detection was $M \geq 0.5$. In contrast to the inverse RMSE map, the highest scores in the PB-OSP score map are consistently high on the true targets.

5.1.4 Noisy Lidar, Noisy Hyperspectral

The previous two sections demonstrate that the proposed fused detection methodology is fairly robust when at least one of the modalities has low noise artifacts. Naturally, there is a point where adding noise to both data sets will cause the method to fail – that is, the true target pixels will no longer have target detection scores that are distinct relative to the background.

5.1.4.1 Detection Results

The noisy spectral and spatial data sets described in Sections 5.1.2 and 5.1.3, respectively, were used to generate the following target detection maps in Figure 5.26. As expected,



Figure 5.26: Detection score maps for noisy Lidar and noisy HSI cube. (a) Inverse RMSE (b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.

these detection maps indicate that when both sensor modalities have significant noise levels present, then the fused detection method is prone to failure. While many of the true target pixels still have high detection scores, the same also holds true for many of the false alarm pixels, such that thresholding will still result in the retention of the false target locations.

5.1.5 Ideal Lidar, Wide Angle Ideal Hyperspectral

5.1.5.1 Map Creation Results

In the event that the Lidar and HSI sensors are not located on a common platform, the Lidar data may still be projected into the frame of the HSI sensor. Again, this is possible only when both sensors are registered to a common geographic coordinate system. In designing a realistic experiment to test the multiple view angle experiment, a Lidar point cloud was collected from a nadir viewing geometry, while an HSI system collected from the same altitude but at a 30° oblique viewing angle.

Since the HSI sensor has a longer slant path to the scene center, the GIFOV of the sensor increased correspondingly. The Lidar data set was designed to span the entire FOV of an HSI sensor located at nadir; hence, after moving the HSI sensor to the oblique viewing geometry, the Lidar cloud no longer fully filled the HSI FOV. This is evident in the lack of feature information along the vertical edges of the feature maps. A sparse z-buffering of the point cloud, as described in Section 4.1.5, was not applied prior to the creation of the feature maps since the change in HSI viewing geometry did not change target occlusion levels significantly.



Figure 5.27: Feature maps for ideal data sets and wide angle HSI sensor. (a) Shadow map (b) incident illumination map (c) shape factor map and (d) pixel purity map.

5.1.5.2 Detection Results

Figure 5.28 demonstrates that the detection process still works relatively well when the Lidar and HSI sensors are not co-located. Both detection maps generally have high detection



Figure 5.28: Detection score maps for ideal data sets and wide angle HSI sensor. (a) Inverse RMSE, (b) PB-OSP, and (c) Truth map for resolved (green) and subpixel (red) targets.

scores across the true targets; however, the PB-OSP score map appears to yield greater suppression of false alarms.

5.2 Real-world Data

As of writing this document, real-world multi-modal data sets are not available for evaluating the fused approach. However, real Lidar data sets were available for estimating the utility in the point cloud processing and feature map techniques described in Sections 4.1 and 4.2, respectively. In the fall of 2006, the Leica Corporation overflew the RIT campus, to include the Microscene I area, with an ALS 50 instrument. The ALS 50 is a line scanned topographic Lidar system, operating on a laser line center of 1.064 μm with a pulse width of 0.5 ns. At the time of the collect, the microscene area contained one vehicle inclined on the archery berm. The resolution of the resulting processed point cloud was approximately 0.4 m in the direction of flight and 0.25 m along the scan direction. Figure 5.29 shows a screen capture of the Leica data set for Microscene I and an image of the target vehicle within the scene. The visualization of the point cloud was produced using the Merrick Advanced



Figure 5.29: (a) Range image of Leica point cloud (b) Image of pickup truck within Microscene I.

Remote Sensing (MARS[®]) software utility [Merrick and Company 2006].

5.2.1 Point Cloud Processing Results

Application of the spin-image target detection techniques on the Leica data proved to be an interesting challenge. Not only did the collect provide data with true noise artifacts, but also the sampling density on the pickup was extremely limited. Inspection of the data cloud revealed that approximately 30 point samples actually fell on the truck; hence, estimation of accurate surface normals would be difficult at best. Leica applied proprietary preprocessing techniques to their raw point clouds in order to eliminate false returns and geo-register the points. Furthermore, in delivering the processed clouds, Leica also supplied extracted, noninterpolated ground plane points for the clouds. These ground points were used to apply a z-window to the data, the results of which are shown in Figure 5.30 below. Thresholds of 0.5



Figure 5.30: Target point extraction results: z-windowed Leica point cloud.

m and 2.5 m were used to define the extent of the z-window. Next, the remaining z-windowed points were filtered using the spin-image detection algorithm based on the following spin-

image generation criteria:

- Spin support = 5.0 m
- Bin size = 0.3 m
- Spin angle = 30°



Figure 5.31: Target point extraction results: scene correspondences (yellow) and extracted target points (red) for Leica point cloud.

Similar to the spin-image detection results for the simulated noisy Lidar point cloud in Figure 5.31, the majority of the z-windowed points are labeled as potential target points.

However, it is interesting to note that all of the samples from the truck resulted in valid correspondence points and the second highest ranked correspondence for the entire scene was a true scene correspondence located on the hood of the pickup truck. Given the number of correspondences in the point cloud, the high scoring correspondence on the truck is not likely to be coincidental. Recall that a correspondence is a point that (1) has a spin-image with a high similarity match score relative to the model spin-image library and (2) passes the geometric consistency filter. Clearly, the result in Figure 5.31 has a significant number of false correspondences. While these output correspondences could be thresholded according to their respective geometric consistency scores, a logical method for determining an appropriate threshold value proved to be elusive. Hence, all correspondences were retained in hopes of suppressing false alarms via subsequent spectral methods.

5.2.2 Map Creation Results

All point cloud processing techniques were applied to the Leica cloud in order to evaluate their usefulness in estimating features in real data. Since no true spectral data was available at the time of the Leica Lidar overflight, a notional spectral sensor was created for the purpose of projecting the point cloud feature data onto an HSI FPA. The hypothetical HSI sensor was pointed at the scene center and designed to have a GIFOV of 1.0 m when oriented at nadir. The feature maps derived from the Leica data are presented in Figure 5.32. Unfortunately, since no spectral data was available during the flight, evaluation of the feature map content is problematic. As an alternative to observing a false color composite of the scene, a range map was generated from the point cloud itself, as is illustrated in Figure 5.33. With knowledge of the sun's position and general height information for scene features,



(c)

(d)

Figure 5.32: Feature maps for Leica point cloud and notional HSI sensor. (a) Shadow map (b) incident illumination map (c) shape factor map and (d) pixel purity map.

then a qualitative assessment of shadow positions and general shape factor levels is possible. As such, the gray levels across the shadow map in Figure 5.32(a) and the shape factor map in Figure 5.32(c) appear to be reasonable. After interpolating the supplied ground plane points to a 0.5 *m* grid, the incident illumination angle map in Figure 5.32(b) was generated. Despite the low sampling densities across the original point cloud, the illumination angle map still appears to be a reasonable estimation of ground plane orientation relative to the solar direction. The location of the archery berm is clearly visible in the map, while flat terrain areas are relatively constant. The effect of averaging many point normals for every HSI pixel serves to smooth outliers in the point normal estimates.



Figure 5.33: Range map for Leica point cloud.

5.3 Conclusions

This research demonstrates that the performance of physics-based, spectral target detection methods may be improved through the incorporation of spatial information. Naturally, the amount of improvement is largely dependent upon the resolution and accuracy of the spatial data. Various methods were used to process Lidar point clouds for spatial features of interest. Methods based on sparse ray tracing techniques proved to be more robust than those that were dependent upon the estimation of local point normals. Since the sparse shadow and skydome feeler algorithms rely solely on ray/volume intersections, the impact of point location error on these algorithms' outputs is not severe. In contrast, point location accuracy is critical when characterizing the spatial variability in a small number of points, as is done when calculating a point normal vector based on eigenvector decomposition. Even with excellent knowledge of a Lidar sensor's position and pointing geometry, range quantization within a point cloud can alter the estimation of point normals and ultimately degrade the performance of the spin-image target detection algorithm.

Based on the documented data sets, one may draw loose conclusions about the Lidar system specifications required to improve hyperspectral detection of vehicles. Lidar data clearly provides several valuable pieces of information to the described fused approach, with possibly the two most important being shadow and target cues. First, the locations and depths of shadows within a scene are critical for predicting accurate sensor-reaching target signatures. A system with mean post spacings on the order of 0.3 m should provide adequate sampling resolution for detecting large shadowed regions. Obviously, the required post spacing will decrease as the need for fine-scale shadow detection increases. Additionally, smaller

5.3. CONCLUSIONS

post spacings are necessary for robust vehicle detection based on the spin-image matching algorithm [Johnson 1997; Vasile and Marino 2004]. Again, Johnson and Vasile claim that at least 200 samples on a target are required for robust detection performance based on Lidar information alone. Given that the exposed area of most common vehicles ranges from 7 to 9 m^2 (e.g., the area of a Toyota Tundra pickup truck when viewed from nadir is approximately 8.7 m^2), 0.3 m posts should provide a modest number of samples on a target vehicle that is well below 200. This research has demonstrated that spin-image based filtering may still be successfully applied to such low density point clouds, albeit with increased false alarm levels. The number of false alarms is largely dependent upon *both* the transverse and range resolutions of a point cloud. In order to achieve accurate spin-image matching, the range resolution of a system should be significantly higher than the transverse. A suggested value of 0.5 ns timing accuracy results in a range resolution of 0.075 m or 4× spatial oversampling in the range dimension compared to the transverse post spacing.

Data level fusion of spatial and spectral data was achieved by projecting 3D points and their respective spatial features into the 2D indexed coordinate system of an HSI sensor. Such fusion of spatial information allows a complex radiometric model to be constrained on a localized basis and potentially results in the creation of optimized signature spaces for every HSI pixel location. Compared to traditional physics-based target detection approaches, the resulting target spaces from the described fused methodology have a reduced span and ultimately allow for improved discrimination for challenging target pixels. Due to the adaptive nature of these target spaces, spectral detection of targets under a broad range of illumination, orientation, and concealment conditions may be realized.

Chapter 6

Future Work

6.1 Additional analysis

Given the sheer number degrees of freedom in the proposed target detection methodology, a rigorous analysis of the impact of each process variable upon the final detection map becomes a daunting task. While the results presented in Chapter 5 represent a relevant set of system scenarios, many additional experiments are necessary in order to truly assess the strengths and weaknesses of the fused approach.

One such experiment that would be particularly useful is a thorough assessment of the degree of spatial oversampling required to effectively constrain a physics-based model. The presented results demonstrate that $25 \times$ spatial oversampling of the Lidar relative to an HSI GIFOV does produce quality fused target detection maps. While the usefulness of spatial data will obviously degrade with the level of oversampling, additional studies to quantitatively characterize such degradation would be informative.

While all of the experiments maintain a GSD of approximately 1.0 *m* for the hyperspectral sensor, running a scenario with larger pixels would be worthwhile. The use of larger pixels increases the likelihood of a non-homogenous background contribution for a given mixed pixel. While mixing in the radiance domain based on an empirical estimate of the background does allow for some flexibility in the degree of background homogeneity, providing an accurate estimate of the local background radiance may become increasingly problematic as the HSI pixel size grows.

An additional area warranting further analysis is the implementation of spin-image based target detection. Several publications on applying spin-image detection to topographic Lidar data sets are available in the literature [Vasile and Marino 2004]; however, application of the process to a variety of scenes with varying levels of post-spacing, point location noise, scene clutter, etc. has yet to be presented. In general all published work has been on extremely high resolution point clouds (*i.e.*, post spacings on the order of 10 cm) with very low noise levels. DIRSIG's Lidar simulation utility provides an effective means to create realistic point clouds and perform system level trade studies across a variety of scenes. An area of research that would be relevant for applying spin-image matching techniques to topographic Lidar data is a study of how target model cloud resolution affects matching performance. A high resolution model may actually degrade matching performance due to the fact that fine-scale surface features may be accurately captured during the normal estimation process. An example of this would be curvature along the roof or hood of a vehicle, such as the one pictured in Figure 7.6. Quantization and reduced resolution in the scene point cloud is likely to remove fine-scale features resulting in a rotation of the scene points' normals for the areas in question, relative to the normals derived from a high resolution cloud. Therefore, matching of the model point cloud resolution to that of the scene may be warranted for the purpose of increasing normal vector orientation consistency between model/scene correspondence locations.

One of the disadvantages of the proposed target extraction process described in Section 4.1.4.3 is that the approach results in a significant number of false alarm target points. Again, this approach extracts target points by tracking the scene points that map to common bins between model and scene spin-images for all scene correspondence locations. An alternative

to the proposed approach is to actually fit the model point cloud to the scene point cloud. Several algorithms, to include Iterative Closest Point (ICP) and RANdom SAmple Consensus (RANSAC), could be used for this purpose [Johnson 1997]. Both algorithms are capable of transforming partially overlapping point sets to a common reference frame. ICP may prove to be more effective when very few outlier correspondences are present in the scene point cloud, while RANSAC is designed to perform well in the presence of false correspondences [Zhang 1994; Hartley and Zisserman 2003]. The benefit of using these algorithms lies in the potential for reduction of false alarm target points, particularly so for areas in close proximity to a high density of true scene correspondences. However, the increase in run times associated with implementing these algorithms should also be considered when evaluating their performance. RANSAC in particular is noted for producing suboptimal surface matches when an upper bound is placed on the algorithm's run time.

6.2 Computational Efficiency

One area with significant potential for improvement is the computational efficiency of many of the processes. In particular, several of the processes such as downsampling of a Lidar point cloud or any of the various processes involving point normal vectors call for the sorting of 3D points based on Euclidean distance. Currently, such searches are performed via a brute force FOR-loop, allowing for proper code function albeit with slow speeds. These codes could be optimized considerably by replacing such loop structures with a binary space-partioning tree search logic [Schneider and Eberly 2003]. These logic structures are commonly used to parse points within 3D space via a series of deterministic binary, planar divisions according
to

$$\vec{n} \cdot X - c \stackrel{1}{\underset{\leq}{\leq}} 0 \quad , \tag{6.1}$$

where \vec{n} and c define the equation of a unique plane in 3D space. After the initial division of the point cloud into two half-spaces, each node in the resulting binary tree corresponds to the binary classification of a subsequent subspace. As such, binary tree search methodologies provide an efficient means to identify a more manageable, localized set of points.

An additional area for potential improvement is in the spin-image target detection process. Recall one of the first steps in the spin-image matching process described in Section 4.1.4.2 is the comparison of every scene-based spin-image to every spin-image in a model library. Johnson proposes that the matching process may be accelerated by compressing the spin-image model library using an approach based on principle components analysis [Johnson 1997]. All of the model spin-images are reformed into *n*-dimensional vectors, where *n* is equal to the number of rows times the number of columns in the original spin-image. The model vector space forms a distribution in *n* space which is then characterized according to the primary directions of variability. While the method does implicitly assume that the data is multivariate normal, the process is still valuable for determining which vector directions are most important for describing the vector space. Johnson suggests a method based on correlation for selecting *k* model space eigenvectors which allows for accurate reconstruction of a model spin-image. Finally, all model spin-images are projected into the *k*-dimensional eigenspace where $k \ll n$. In order to match scene-based spin-images to the compressed library, the scene images are mean subtracted and subsequently projected onto the k eigenvectors describing the model library vector space. Since the computational efficiency of binary space-partioning search trees degrades with the dimensionality of the data, a closest point search algorithm by Nene and Nayar is suggested as an optimal search algorithm [Nene and Nayar 1995]. Finally, Johnson reports that the described compression of the spin-image model library results in an order of magnitude decrease in spin-image matching speeds [Johnson 1997].

While the previous paragraphs describe areas that will benefit from increased efficiency, Table 5.4 indicates that most significant computational burden is associated with the various ray tracing techniques. The most primitive implementation of a sparse ray tracer on a cloud consisting of p points calls for p^2 computations. While the current methodology does allow for the elimination of some improbable ray/sphere intersection points, the process could still be improved dramatically. One possible method for increasing the overall speed of the sparse ray tracer is to implement intersection cueing logic based on a low resolution version of the original point cloud. If the point cloud were facetized using Delaunay triangulation, then traditional, optimized ray tracing techniques designed for continuous geometries could be used to cue a sparse ray tracing routine [Schneider and Eberly 2003]. While triangulation of the cloud is likely to result in erroneous spatial structure within the cloud, it does provide a significant reduction in the overall number of required computations and should therefore provide an efficient cue for subsequent processing of localized regions in the sparse cloud. In doing so, the speed of traditional shadow feeler techniques could be leveraged while maintaining the accuracy associated with sparse feeler algorithms.

Chapter 7

Appendix

7.1 Downwelled Path Radiance Estimation

This section describes various methods for estimating the downwelled path radiance vector for a given scene and collection geometry. When creating a simulated HSI cube, DIRSIG's make_adb process estimates the spatially-integrated downwelled path radiance by placing a sensor at the ground and sampling the skydome *multiple times* at uniform intervals of azimuth and zenith. Each sample direction entails a separate MODTRAN run in "slant path to space" mode [Brown 2006]. In contrast, when generating a path radiance triplet for building a target space, a *single* MODTRAN run is used based on the actual collection geometry (*i.e.*, "vertical slant path between two altitudes" mode). The scattered path radiance content for a given scene geometry is largely driven by the implemented atmospheric model (*e.g.*, mid-latitude summer, sub-arctic winter, *etc.*), scattering algorithm (*i.e.*, Isaac's two-stream or DISTORT *n*-stream), and target/background albedo characteristics [Berk *et al.* 2003].

While the sole input to DIRSIG's make_adb process is a cfg file, a tape5 card deck describing the user-defined atmospheric conditions is embedded in the cfg file. For the sake of speed (and at the expense of accuracy), Isaac's two-stream multiple scattering algorithm was used in the DIRSIG tape5 card deck, which is provided in Figure 7.1. A surface albedo of 0.1 was used to approximate the mean albedo of the background throughout the scene. DIRSIG's estimate of the total spatially-integrated downwelled vector for the 23 km visibility atmosphere was parsed from the resulting adb file.

162)										CHA	PTER	₹ 7. ⊥	APPEND	
TS	2	2	2	1	0	0	0	0	0	0	1	0	0	0.000	0.10
F	8F	0	0.00	000		0		0 F	FF		0.	000			
	1	0	0	3	0	0	0.	000	0.	000	0.	000	0	.000	0.000
	12.	500	0.	000	180.	000	0.	000	0.	000	0.	000	0		0.000
	1	0	176	0											
	43.	130	77.	621	0.	000	0.	000	15.	000	0.	000	0	.000	0.000
	0.3 0	373	2.	503	0.	010	0.	010 M	[М	А				

Figure 7.1: DIRSIG tape5 card deck.

For comparison with DIRSIG's output, two different MODTRAN approaches for estimating the downwelled component were accomplished. These approaches are similar to what would be used to estimate the path radiances for building a target space. The first approach represents the downwelled estimate using the two-run method described in Section 4.3. The tape5 file for estimating the ground-leaving path radiances using the two-run method is shown below in Figure 7.2. Again, the two-run method describes *both* the target and surrounding background as a spectrally constant, 100% Lambertian reflector when estimating ground-leaving path radiances.

TS	2	2	2	-1	0	0	0	0	0	0	1	0	0	0.000	1.00
Т	8T	0	360.00	0000		0		0 F	FF		0.	000			
	1	1	0	0	0	0	23	.000	0.0	000	0.	000	0.	.000	0.158
	12.	658	0	. 158	180	.000	12.	500	0.0	000	0.	000	0		0.000
	1	2	176	0											
	43.	130	77	.621	0	.000	0.	.000	15.0	000	0.	000	0.	.000	0.000
	0.	373	2	. 503	0	.010	0.	010RM		MR	А				
	0														

Figure 7.2: Tape5 card deck for two-run method.

The second approach was based on the single-run method where the target and background reflectance spectra were defined as 100% and 0% constant Lambertian reflectors, respectively. The associated card is provided in Figure 7.3. Note that both of the singleand two-run card decks simulate multiple scattering using the DISTORT 8-stream model.

2	2	2	-1	0	0	0	0	0	0	1		0	0	0.000	LAMBER
8T	0	360.00	0000		0		0	FFF			0.00	0			
1	0	0	3	0	0	0	.000	0	.000		0.00	0	0.	000	0.158
12.6	58	0.	158	180.	000	12	.500	0	.000		0.00	0	0		0.000
1	2	176	0												
43.1	.30	77.	621	0.	000	0	.000	15	.000		0.00	0	0.	000	0.000
0.3	373	2.	503	0.	010	0	.010R	М	М	R A					
300.0	000														
pec_al	b_r	new.dat	;												
consta	nt,	100%													
consta	nt,	0%													
0															
	2 8T 1 12.6 43.1 0.3 300.0 pec_al consta 0	2 2 8T 0 1 0 12.658 1 2 43.130 0.373 300.000 pec_alb_r constant, 0	2 2 2 8T 0 360.00 1 0 0 12.658 0. 1 2 176 43.130 77. 0.373 2. 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 8T 0 360.00000 1 0 0 3 12.658 0.158 1 2 176 0 43.130 77.621 0.373 2.503 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 8T 0 360.00000 1 0 0 3 0 12.658 0.158 180. 1 2 176 0 43.130 77.621 0. 0.373 2.503 0. 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 8T 0 360.00000 0 1 0 0 3 0 0 12.658 0.158 180.000 1 2 176 0 43.130 77.621 0.000 0.373 2.503 0.010 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 8T 0 360.00000 0 1 0 0 3 0 0 0 12.658 0.158 180.000 12 1 2 176 0 43.130 77.621 0.000 0 0.373 2.503 0.010 0 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 8T 0 360.00000 0 0 1 0 0 3 0 0 0.000 12.658 0.158 180.000 12.500 1 2 176 0 43.130 77.621 0.000 0.000 0.373 2.503 0.010 0.010R 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 8T 0 360.00000 0 0 F F F 1 0 0 3 0 0 0.000 0 12.658 0.158 180.000 12.500 0 1 2 176 0 43.130 77.621 0.000 0.000 15 0.373 2.503 0.010 0.010RM 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 8T 0 360.00000 0 0 F F F 1 0 0 3 0 0 0.000 0.000 12.658 0.158 180.000 12.500 0.000 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.373 2.503 0.010 0.010RM M 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 1 8T 0 360.00000 0 0 FFF 1 0 0 3 0 0 0.000 0.000 12.658 0.158 180.000 12.500 0.000 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.373 2.503 0.010 0.010RM MR A 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 1 8T 0 360.00000 0 0 FFF 0.00 1 0 0 3 0 0 0.000 0.000 0.00 12.658 0.158 180.000 12.500 0.000 0.00 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.00 0.373 2.503 0.010 0.010RM MR A 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 1 0 8T 0 360.00000 0 0 FFF 0.000 1 0 0 3 0 0 0.000 0.000 0.000 12.658 0.158 180.000 12.500 0.000 0.000 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.000 0.373 2.503 0.010 0.010RM MR A 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 1 0 0 8T 0 360.00000 0 0 FFF 0.000 1 0 0 3 0 0 0.000 0.000 0.000 0. 12.658 0.158 180.000 12.500 0.000 0.000 0 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.000 0. 0.373 2.503 0.010 0.010RM MR A 300.000 pec_alb_new.dat constant, 100% constant, 0% 0	2 2 2 -1 0 0 0 0 0 0 1 0 0 0.000 8T 0 360.00000 0 0 FFF 0.000 1 0 0 3 0 0 0.000 0.000 0.000 0.000 12.658 0.158 180.000 12.500 0.000 0.000 0 1 2 176 0 43.130 77.621 0.000 0.000 15.000 0.000 0.000 0.373 2.503 0.010 0.010RM MR A 300.000 pec_alb_new.dat constant, 100% constant, 0% 0

Figure 7.3: Tape5 card deck for single-run method.

Additional differences between the DIRSIG and target space tape5 card decks include the scattering reference location (DIRSIG IMULT = 1, target space IMULT = -1), CO₂ mixing ratio (DIRSIG CO2MX = 330 *ppmv*, target space CO2MX = 360 *ppmv*), and aerosol phase function (DIRSIG IPH = Henyey-Greenstein, target space IPH = Mie-generated). Note that several other fields in the card decks are different. However, such discrepancies are resolved by one of two means. First, the differences between the H1, H2, and GNDALT fields are resolved by DIRSIG. Prior to running MODTRAN, make_adb parses the tape5 card deck within the scene cfg file to ensure that the correct scene geometry is used. Secondly, several of the flags may appear to be different (*e.g.*, VIS and IHAZE); however, these differences are attributed to hard coding a default value versus simply allowing MODTRAN to run using its defaults.

Finally, Figure 7.4 shows a comparison of the various downwelled radiance estimates.

Despite the aforementioned differences between the tape5 card decks, the curves line up well in general. The single- and two-run methods for running MODTRAN represent extrema for background adjacency contribution to the downwelled path radiance vector. Since the two-run method effectively models the background as a 100% Lambertian reflector, the adjacency contribution is maximized. Conversely, the adjacency contribution is eliminated from the single-run method. These adjacency differences are evident in the target space generation downwelled curves, as the two-run downwelled estimate is consistently higher than the single-run, with separation between the curves being proportional to λ^{-4} .



Figure 7.4: Downwelled path radiance estimation comparison.

When comparing the single- and two-run downwelled curves to DIRSIG's, it is interesting to note that an inflection point occurs. Both target space downwelling estimation methods underestimate DIRSIG's downwelled curve at wavelengths below 500 nm, and above which, overestimation occurs. It is important to recognize that the downwelled path estimates and their associated errors (*i.e.* assuming DIRSIG's downwelled estimate as truth) propagate into a target space. Referring back to Figure 5.11(a) and (b), the discrepancy between the best target space vector and the radiance vector for a given target pixel also occurs in the 400-500 nm region. Variability within the target space allows for correction of the overestimation above 500 nm; however, doing so may also result in accentuating the underestimated downwelled contribution in the 400-500 nm range.

As an aside, the 23 km visibility atmosphere used to build the nadir-oriented, synthetic HSI data sets in Chapter 5 was created using the DIRSIG tape5 card deck shown in Figure 7.1. Likewise, the downwelled path radiance estimate used in forming the associated target spaces was calculated using the two-run method card deck in Figure 7.2. This provides insight to the overall strength of the invariant approach. Despite the fact that the atmosphere used to construct the target space did not match the atmosphere of the scene, the target detection approach still performed well.

7.2 Detection Results Using Different Atmospheres

All of the results described in Chapter 5 were produced using ideal atmospheric conditions. Specifically, MODTRAN was used to generate an atmosphere with 23 km visibility in a rural setting. This section describes the impact of reducing atmospheric visibility upon the fused target detection process.

Referring to Figure 5.9, the number of spatial maps that was used to created the con-

strained signature spaces was largely driven by the relative importance of the path radiance terms in Eq. (4.31). Note that the path radiance terms for a 23 km visibility atmosphere shown in Figure 3.1(a) are vastly different from the corresponding curves for 12 and 5 km atmospheres shown in Figure 7.5. A decrease in visibility is largely due to more scatter-



Figure 7.5: Path radiance curves for different atmospheres. (a) $12 \ km$ and (b) $5 \ km$.

ing events in the atmosphere and causes the magnitude of the upwelled term to increase correspondingly, particularly in the blue regime. These path radiance curves indicate that the relative importance of each path radiance term will change with visibility levels, as will the importance of including the proper amount of variability when creating a constrained signature space.

Based on the curves in Figure 7.5, 5 shadow (K), 3 incident illumination angle (θ) , 1 shape factor (F), and 5 pixel purity (M) maps were used to build the space for the 12 km visibility atmosphere. This particular map configuration was chosen because the magnitude of the downwelled term was still less than that of the direct solar term, placing more of an accuracy burden on the K and θ terms when generating signature spaces. In contrast, the magnitudes of the same terms are nearly equivalent for the 5 km visibility atmosphere. As a result, 3 shadow, 3 incident illumination angle, 3 shape factor, and 3 pixel purity maps were used to build the space for the 5 km visibility atmosphere.

12 km Visibility

Based on the new set of spatial feature maps that were optimized for a $12 \ km$ atmosphere, a new target space was created, after which the PB-OSP detection statistic was applied to the new scene. Similar to the results presented in Section 5.1.1.4, thresholds were applied to detection score maps to produce Tables 7.1-7.3.

- Fused subpixel detection threshold level = 0.08
- Fused resolved detection threshold level = 0.08
- HSI only subpixel detection threshold level = 0.20
- HSI only resolved detection threshold level = 0.225

	TRUTH		DETEC	CTIONS	DETECTIONS HSI ONLY		
			FUS.	ED			
Target #	Mixed	Pure	Mixed	Pure	Mixed	Pure	
3	9	4	5	4	6	4	
7	8	2	1	1	0	0	
8	11	4	2	1	3	3	
Totals	28	10	8	6	9	7	

Table 7.1: Pixel-level PB-OSP performance comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using $12 \ km$ visibility atmosphere.

METHOD	THRESHOLD LEVEL	# OF FALSE ALARMS	
Fused	Mixed	13	
Fused	Pure	13	
HSI only	Mixed	32	
HSI only	Pure	28	

Table 7.2: Pixel-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using 12 km visibility atmosphere.

	FALSE	DETECTED	DETECTED
	ALARM	FUSED	HSI ONLY
Target 1	Y	Y	Y
Target 2	Y	N	Y
Target 3	N	Y	Y
Target 4	Y	Y	N
Target 5	Y	N	N
Target 6	Y	N	N
Target 7	N	Y	N
Target 8	N	Y	Y
Target 9	Y	Y	Y

Table 7.3: Object-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using $12 \ km$ visibility atmosphere (green = correct, red = incorrect).

5 km Visibility

Tables 7.4-7.6 show the detection performance comparison for the 5 km visibility data set.

- Fused subpixel detection threshold level = 0.115
- Fused resolved detection threshold level = 0.115
- HSI only subpixel detection threshold level = 0.15
- HSI only resolved detection threshold level = 0.175

	TRUTH		DETEC	CTIONS	DETECTIONS HSI ONLY			
			FUS.	ED				
Target $\#$	Mixed	Pure	Mixed	Pure	Mixed	Pure		
3	9	4	5	3	5	4		
7	8	2	2	0	0	0		
8	11	4	2	4	3	3		
Totals	28	10	9	7	8	7		

Table 7.4: Pixel-level PB-OSP performance comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using $5 \ km$ visibility atmosphere.

METHOD	THRESHOLD LEVEL	# OF FALSE ALARMS		
Fused	Mixed	30		
Fused	Pure	30 53		
HSI only	Mixed			
HSI only	Pure	20		

Table 7.5: Pixel-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using $5 \ km$ visibility atmosphere.

	FALSE	DETECTED	DETECTED
	ALARM	FUSED	HSI ONLY
Target 1	Y	Y	Y
Target 2	Y	N	Y
Target 3	N	Y	Y
Target 4	Y	Y	N
Target 5	Y	N	N
Target 6	Y	Y	N
Target 7	N	Y	N
Target 8	N	Y	Y
Target 9	Y	Y	Y

Table 7.6: Object-level PB-OSP false alarm comparison between fused approach and Ientilucci's method (HSI only), based on ideal Lidar/HSI data using 5 km visibility atmosphere (green = correct, red = incorrect).

7.3 Impact of Range Quantization on Spin-Images

This section serves as a preliminary study on the impact of range quantization on spinimages generated from topographic Lidar point clouds. Since quantization is likely to alter the locations of cloud points, it can have a significant impact upon the calculation of local point normals. Furthermore, since a point's normal vector is used to define the local basis for the point's corresponding spin-image, quantization errors may result in the rotation of a spin-image basis.

7.3.1 Quantized Point Clouds

Point clouds with 10.0 *cm* mean post spacings were generated from the 3D CAD model shown in Figure 7.6 below. These clouds were created using DIRSIG's Lidar simulation tool and exhibit varying levels of range quantization. Furthermore, the Lidar illumination angle



Figure 7.6: CAD model for SUV.

of the CAD model was varied in order to alter the direction of the range dimension for the point clouds.

A total of six quantized point clouds were generated. Three of the six were generated from a nadir Lidar illumination angle with temporal range bins of 0.5, 1.0, and 2.0 ns, which equate to spatial range (round trip) bins of approximately 30.0, 60.0, and 120.0 cm. The remaining three quantized point clouds were generated from a 45° illumination angle based on the same quantization levels. 2D cross sections of the quantized clouds are presented in Figure 7.7. These clouds are false colored according to height along the z-axis. The effect of range quantization relative to the range dimension of the cloud is clear. For the clouds collected from nadir, the planar structure of the surfaces along the hood and roof of the vehicle is preserved, while the vertical spacings between the points change with the level of quantization. In contrast, the clouds collected from a 45° slant angle do not preserve the shape of the hood and roof, while the point-to-point spacing in the z dimension of the data remains fairly constant with differing quantization.





Figure 7.7: Quantized point clouds for the SUV CAD model. (a) Nadir at 0.5 ns (b) slant at 0.5 ns (c) nadir at 1.0 ns (d) slant at 1.0 ns (e) nadir at 2 ns and (f) slant at 2 ns.

7.3.2 Impact on Point Normals

As is illustrated in Figure 7.7, range quantization effectively distorts the spatial shape of the point samples along the model. However, it is important to recognize that the impact of this distortion upon spin-images may be greater than simply skewing the brightness distributions across the spin-image histogram. Recall that point normals are calculated based on eigenvalue decomposition of *n*-closest, non-collinear points. Range quantization effectively forces local point samples that share the same quantization bin to become planar oriented in a direction orthogonal to the range dimension of the point cloud. *Thus, a likely byproduct of range quantization is the common alignment of point normals sharing the same quantization bin.* Figure 7.8 shows the impact of range quantization upon point normals when 3-closest, non-collinear points are used for the calculation of normal vectors.





Figure 7.8: Quantized normals for the SUV CAD model based on n=3. (a) Nadir at 0.5 ns (b) slant at 0.5 ns (c) nadir at 1.0 ns (d) slant at 1.0 ns (e) nadir at 2 ns and (f) slant at 2 ns.

One simple method for mitigating the impact of quantization when calculating a point normal is to increase the number of points n considered when performing a localized eigenanalysis. If n is sufficiently high that points from multiple range bins are considered, then point normals begin to align themselves in along the z-axis of the point cloud, regardless of quantization direction or surface orientation. As is shown in Figure 7.9, increasing n has little to no effect on the normals for the point clouds collected from nadir; however, the change in point normals the clouds collected at a 45° slant angle is significant. This quantization mitigation strategy undoubtedly adds error to some point normals. Clearly, the normals along the windshield regions are incorrect. However, since these errors are predictable then they can be incorporated into the spin-image model library – that is, the user may choose to force normals for all model points to be [0, 0, 1] in an effort to improve spin-image matching. It is important to recognize that accuracy in point normal estimates is not as critical as is consistency between the normals for a given scene and model correspondence location.



Figure 7.9: Quantized normals for the SUV CAD model based on n=6. (a) Slant at 0.5 ns (b) slant at 1.0 ns and (c) slant at 2 ns.

7.3.3 Impact on Spin-images

Since a point's normal vector determines the basis axes from which the point's spinimage is derived, erroneous point normal vector orientations can significantly degrade the performance of spin-image matching techniques. The impact of an error in a point normal upon a subsequent spin-image varies with spin-image bin size and the radial distance of a contributing point from the normal. Points that are relatively close to the spin-image basis point, as measured in bin size units, are likely to map to the same spin-image bin, even with considerable error in the point normal vector. Figure 7.10 illustrates normals and spinimages for three different point clouds, based on roughly the same spin-image basis point location. The generation criteria used in creating the spin-images are as follows:

- Spin support = 5.0 m
- Bin size = 0.25 m
- Spin angle = 180°

While the quantized examples represent the extreme cases of the point clouds presented, they do serve to illustrate the point that range quantization can have a severe impact on one's ability to effectively match based on spin-image techniques.





Figure 7.10: Basis points, normals, and spin-images. (a) Perfect model cloud (b) perfect spin-image (c) nadir at 2.0 ns, n=6 (d) nadir at 2.0 ns, n=6 spin-image (e) Slant at 2 ns, n=6 (f) Slant at 2 ns, n=6 spin-image (g) Slant at 2 ns, n=3 and (h) Slant at 2 ns, n=3 spin-image.

Works Cited

- Baum, Jerrold E. and Steven J. Rak. "Simultaneous active/passive IR vehicle detection." SPIE, 1991, 209–220.
- Brown, Scott D. 2005 "The CDMatrix Class.". Rochester Institute of Technology.
- Brown, Scott D. DIRSIG User's Manual, Release 4. Chester F. Carlson Center for Imaging Science, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY 14623-5604, 2006. http://dirsig.cis.rit.edu/doc/manual/manual.html.
- Carmichael, O. and D. Huber. "Large Data Sets and Confusing Scenes in 3-D Surface Matching and Recognition." Proc. of Second Intl. Conf. on 3-D Imaging and Modeling. Ottawa, CA: , 1999, pp. 358–367.
- Chandra, Kartik and Glenn Healey. "Recovering surface properties for hyperspectral scenes." SPIE, 2006, 62331H.
- Cisz, Adam. Performance Comparison of Hyperspectral Target Detection Algorithms. Master's thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2006.
- Cobzas, F. and H. Zhang. "Planar Patch Extraction with Noisy Depth Data." 2001, pp. 240–245.
- Epema, D.H.J, M. Livny, R. van Dantzig, X. Evers, and J. Pruyne. "A Worldwide Flock of Condors : Load Sharing among Workstation Clusters." *Journal on Future Generations* of Computer Systems 12 (1996).
- Farmer, W.M. The Atmospheric Filter: Volume II Effects. Winter Park, FL: JCD Publishing, 2001.

- Feng, Xiaofan. Design and Performance of a Modular Imaging Spectrometer Instrument. PhD dissertation, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 1995.
- Foley, James, Andries van Dam, Steven Feiner, and John Hughes. Computer Graphics: Principles and Practice. 2nd edition. Reading, Massachusetts: Addison Wesley, 1992.
- Gross, Harry N. and John R. Schott. "Application of spatial resolution enhancement and spectral mixture analysis to hyperspectral images." SPIE, 1996, 30–41.
- Hannon, Stephen M. and Jeffrey H. Shapiro. "Active-passive detection of multipixel targets." SPIE, 1990, 2–23.
- Hartley, R. and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge, UK: Cambridge University Press, 2003.
- Healey, G. and D. Slater. "Models and methods for automated material identification in hyperspectral imagery acquired under unknown illumination and atmospheric conditions." *IEEE Transactions on Geoscience and Remote Sensing* 37 (November 1999): 2706–2717.
- Ientilucci, E.J. "Two Dimensional Decision Spaces Generated from Physics Based Target Detection as Applied to Hyperspectral Imagery." 2006.
- Ientilucci, Emmett J. Hyperspectral Sub-pixel Target Detection Using Hybrid Algorithms and Physics Based Modeling. PhD dissertation, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2005.
- Jet Propulsion Laboratory. http://aviris.jpl.nasa.gov/, AVIRIS website, July 2006.

- Johnson, Andrew E. Spin-images: A Representation for 3-D Surface Matching. PhD dissertation, Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA, 1997.
- Klempner, Scott L., Brent Bartlett, and John R. Schott. "Ground truth-based variability analysis of atmospheric inversion in the presence of clouds." SPIE, 2006, 630109.
- Kostakis, Joseph, Matthew L. Cooper, Jr. Thomas J. Green, Michael I. Miller, Joseph A. O'Sullivan, Jeffrey H. Shapiro, and Donald L. Snyder. "Multispectral sensor fusion for ground-based target orientation estimation: FLIR, ladar, HRR." SPIE, 1999, 14–24.

Lach, Stephen. 2007 "Personal correspondence with Stephen Lach.".

- Lanorte, Vito, Stefano Pignatti, and Valerio Tramutoli. "Extending the use of ATREM atmospheric correction to MIVIS data." SPIE, 2002, 285–289.
- Lee, K. A sub-pixel scale target detection algorithm for hyperspectral imagery. PhD dissertation, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2003.
- Liu, Yong and Glenn Healey. "Recognizing 3D objects in cluttered backgrounds." SPIE, 2003, 9–16.
- Liu, Yong and Glenn Healey. "Discriminant analysis with nonparametric estimates for subpixel detection of 3D objects in hyperspectral imagery." SPIE, 2006, 623304.
- Manolakis, D. and G. Shaw. "Detection algorithms for hyperspectral imaging applications." *IEEE Signal Processing Magazine* 19 (January 2002): 29–43.
- Measures, Raymond M. Laser Remote Sensing: Fundamentals and Applications. New York: John Wiley and Sons, Inc., 1984.

- Merrick and Company. http://www.merrick.com/servicelines/gis/mars.aspx, Merrick and Company website, June 2007.
- Messinger, D.W. Tool #2: Detection of Concealed and Contaminated Targets Algorithm Theory and Tool Usage. RIT NURI: Exploitation Tool Development for Hyperspectral Imagery, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2005.
- Miller, Craig J. "Performance assessment of ACORN atmospheric correction algorithm." SPIE, 2002, 438–449.
- Nene, S. and S. Nayar. A simple algorithm for closest point search in high dimensions. Technical Report CUCS-030-95, Columbia University, October 1995.
- Optech Incorporated. http://optech.ca/pdf/Specs/specs_altm_3100.pdf, Optech Inc. website, 2004.
- Pan, Zhihong and Glenn Healey. "Recognizing 3D objects in hyperspectral images under unknown conditions." SPIE, 2001, 559–569.
- Perkins, Timothy, Steven Adler-Golden, Michael Matthew, Alexander Berk, Gail Anderson, James Gardner, and Gerald Felde. "Retrieval of atmospheric properties from hyper and multispectral imagery with the FLAASH atmospheric correction algorithm." SPIE, 2005, 59790E.
- Peterson, E.D. Validation and Verification of Surface and Buried Landmine Signatures in DIRSIG. Master's thesis, Rochester Institute of Technology, 54 Lomb Memorial Drive, Rochester, NY, 2004.

- Ratches, James A., C.P. Walters, Rudolf G. Buser, and B.D. Guenther. "Aided and automatic target recognition based upon sensory inputs from image forming systems." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 19 (September 1997): 1004–1019.
- Rodvold, David and Tim J. Patterson. "Multisensor image cueing (MUSIC)." SPIE, 2002, 144–151.
- Sadjadi, Firooz A. "Application of genetic algorithm for automatic recognition of partially occluded objects." SPIE, 1994, 428–434.
- Schneider, P.J. and D.H. Eberly. Geometric Tools for Computer Graphics. San Francisco, CA: Morgan Kaufman Publishers, 2003.
- Schott, J.R. Remote Sensing: The Imaging Chain Approach. New York: Oxford University Press, 1997.
- Schott, J.R. Remote Sensing: The Imaging Chain Approach. 2nd edition edition. New York: Oxford University Press, 2007.
- Sithole, George. Segmentation and Classification of Airborne Laser Scanner Data. PhD dissertation, Netherlands Geodetic Commission, Royal Netherlands Academy of Arts and Sciences, P.O. Box 5058, 2600 GB Delft, The Netherlands, 2005.
- Snorrason, Magnus, Harald Ruda, and Alper Caglayan. "Automatic target recognition in laser radar imagery." Acoustics, Speech, and Signal Processing, International Conference on. MAY 1995, 2471–2474.
- Stein, F. and G. Medioni. "Structural indexing: efficient 3-D object recognition." Pattern

Analysis and Machine Intelligence, IEEE Transactions on 14 (February 1992): 125–145.

- Thai, B. and G. Healey. "Invariant subpixel material detection in hyperspectral imagery." *Geoscience and Remote Sensing, IEEE Transactions on* 40 (March 2002): 599–608.
- U.S. Geological Survey. http://craterlake.wr.usgs.gov/dem.html/, U.S. Geological Survey website, June 2006.
- Vasile, Alexandru N. and Richard Marino. "Pose-independent automatic target detection and recognition using 3D LADAR data." SPIE, 2004, 67–83.
- Vosselman, G. "Slope Based Filtering of Laser Altimetry Data." IAPRS. Vol 33, Amsterdam: 2000.
- Wellfare, Michael R. and Karen Norris-Zachery. "Characterization of articulated vehicles using ladar seekers." SPIE, 1997, 244–254.
- Zhang, Z. "Iterative point matching for registration of free-form curves and surfaces." Journal of Computer Vision 13 (1994): 119–152.
- Zheng, Q., S.Z. Der, and H.I. Mahmoud. "Model-based target recognition in pulsed ladar imagery." *Image Processing, IEEE Transactions on* 10 (April 2001): 565–572.
- Zhou, Yi-Tong and Demetrios Sapounas. "IR/ladar automatic object recognition system." SPIE, 1997, 119–128.