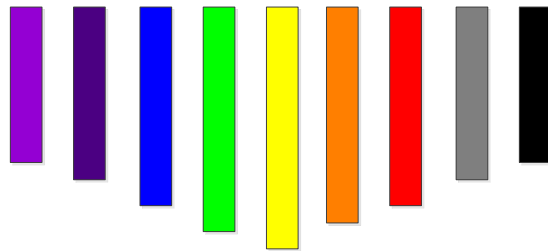
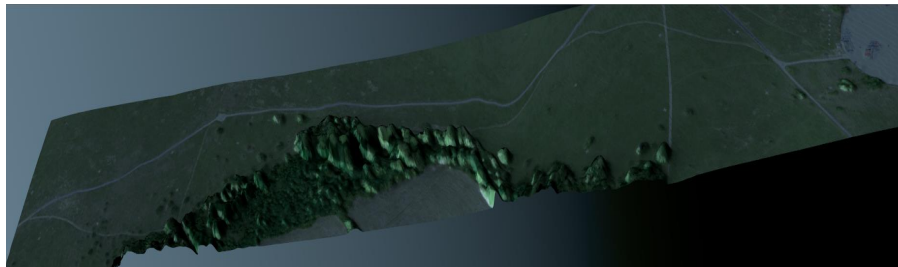

CHIMES



Cranfield Hyperspectral Image Modelling and Evaluation System



Usman A. Zahidi

CRANFIELD UNIVERSITY

USMAN A. ZAHIDI

CHIMES: AN ENHANCED END-TO-END CRANFIELD
HYPERSPSPECTRAL IMAGE MODELLING AND
EVALUATION SYSTEM

CENTRE FOR ELECTRONIC WARFARE
Ph.D. Dissertation

PhD in Defence and Security
Academic Year: 2016–2020

Supervisors: Dr. Peter Yuen, Dr. David James
July 2016 - February 2020

CRANFIELD UNIVERSITY

CENTRE FOR ELECTRONIC WARFARE
Ph.D. Dissertation

PhD in Defence and Security

Academic Year: 2016–2020

USMAN A. ZAHIDI

CHIMES: An Enhanced End-to-End Cranfield Hyperspectral
Image Modelling and Evaluation System

Supervisors: Dr. Peter Yuen, Dr. David James
July 2016 - February 2020

This research report is submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy.

© Cranfield University 2020. All rights reserved. No part of
this publication may be reproduced without the written
permission of the copyright owner.

Abstract

Hyperspectral remote sensing enables establishing semantics from an image by providing spectral details used for differentiating materials. The airborne/satellite setup for remote sensing are typically expensive in terms of time and cost-effectiveness. It is therefore important to predict performance of such systems as a precursor. Hyperspectral scene simulation is a technique which allows the detailed spatial and spectral information of a natural scene to be reconstructed without the need for expensive and time-consuming airborne/spaceborne image acquisition systems. It helps in predicting the potential performance of airborne/satellite systems, moreover, it enables varying atmospheric conditions, estimating degradation in sensor performance to provide better uncertainty analysis and traceability, performance analysis of data processing algorithms and counter-measures/camouflage assessment in military applications. Digital Imaging Remote Sensing Image Generation (DIRSIG) developed by Rochester Institute of Technology and Camouflage Electro-Optic Simulator (CameoSim) by Lockheed Martin are the two earliest research and commercial products, respectively, that incorporate hyperspectral rendering for accurate physics-based radiance estimation. Although CameoSim is a well-established Scene simulator, however it does not support volumetric scattering and localised adjacency model. DIRSIG has provided support form these features in newly developed version called DIRSIG5. Due to export control restriction it is typically not possible to access these simulators, hence motivates development of inhouse scene simulator. This thesis summarises the research which constitutes part of the deliverable under the DSTL R-Cloud project for the establishment of an in-house HSI scene simulator, which is known as the Cranfield Hyperspectral Image Modelling and Evaluation System (CHIMES). CHIMES is a physics-based rendering enabled simulator and the main concept follows directly the radiative transfer (RT) big equation, with some components adopted from DIRSIG and CameoSim etc. The goal of the present research has been set and the work has been progressed in the following manner:

- The establishment of CHIMES from scratch;
- Validation of CHIMES through direct comparison with commercial-off-the-shelf

(COTS) simulator such as CameoSim (CS);

- Enhancement of CHIMES over the COTS simulator (e.g. CS) to include automatic in-scene atmospheric parametrisation, localised adjacency-effect model and volumetric scattering to achieve a more realistic scene simulation particularly for the rugged terrain;
- To propose methods on how difficult issues such as shadows can be mitigated in scene simulation.

This thesis summarises the work performed as according to the above 4 objectives with main results as follows:

- CHIMES has been shown to reproduce the scene simulation performed by a COTS simulator (e.g. CameoSim) under various atmospheric conditions.
- An automatic atmosphere parameterisation search algorithm has been shown to be effective to allow the simulation of the scene without the need of repeated trial and error atmospheric parameter adjustments.
- Two adjacency models: the Background One-Spectra Adjacency Effect Model (BOAEM) and the Texture-Spectra Incorporated Adjacency Effect Model (TIAEM) have been developed under this work. The BOAEM is somewhat similar to that adopted in CS with a distinctive feature of volumetric scattering, however, the TIAEM is a terrain dependence adjacency which is much more sophisticated. It has been shown that at high altitude scene, TIAEM performs better than the BOAEM by 6.0% and by 10.0% better than CameoSim particularly in the 2D geometric simulation, in terms of ℓ_1 -norm error. In the lower altitude scene, BOAEM performs better than both TIAEM and CameoSim by 22.0% and 16%. In a 3D scene (i.e. terrain with Digital Elevation Model (DEM)) with sensor at lower altitude CameoSim error raises by 5 times compared to GT. BOAEM still performs better than TIAEM by a similar 22% ℓ_1 -norm error.
- A means for assessing the shadowed pixels of the scene has been proposed and the validation of the model requires more comprehensive ground truth (GT) data which will be performed in the future research.

- Most of the above results have been published in three journal papers as part of the contributions towards the HSI research community.

Table of Contents

Abstract	vi
List of Figures	xii
List of Tables	xviii
List of Abbreviations	xix
Acknowledgements	xxi
1 Introduction	2
1.1 Hyperspectral Imaging	2
1.2 Scene Simulation	3
1.3 Motivation of Research	5
1.4 Problem Statment	6
1.5 Scope and New aspect of work	6
1.6 Thesis Organisation	9
2 Fundamentals of HSI Scene Simulation and RT	10
2.1 Components of Hyperspectral Remote Sensing	12
2.2 Hyperspectral Scene Simulation	13
2.3 Drawbacks in existing simulators	19
2.4 Radiative Transfer	19
2.5 Light Material Interaction	20
2.6 Radiometry	29
2.7 At-sensor Radiance Models and Components	35
2.8 Monte-Carlo Integration	42
2.9 Adjacency Effect Models	57

2.10	Path Tracing vs Ray Tracing	62
3	CHIMES Simulator	67
3.1	Proposed CHIMES Simulator	67
3.2	Components of CHIMES	70
3.3	Atmosphere and Emitter Construction	73
3.4	Spectral Rendering	79
3.5	Radiometric Integrator	82
3.6	Automatic Atmospheric Search for Atmospheric Parameters for Scene Simulation	93
4	Shadow Detection	98
4.1	Shadow Detection in an Hyperspectral image	98
4.2	Radiative Transfer Model-Based Relationship between Shadowed and Non- Shadowed Regions	100
4.3	Proposed Multi-Layered Regression Learning Algorithm	102
5	Experimental Data and Assessment Methodology	114
5.1	Experimental Data	114
5.2	Experimental Setup for Atmospheric Parameters Search	116
5.3	Validation Data for Shadow Detection	117
5.4	Validation of Simulated Images	118
6	Results and Validation	122
6.1	Results and Validation	122
6.2	Simulation Results	122
6.3	Modesto Simulation	138
6.4	Shadow Detection Results	143
7	Discussion and Drawbacks	149
7.1	Scene Simulation	149
7.2	Shadow Detection	151
8	Conclusion And Future Work	153
8.1	Achievements	153
8.2	CHIMES Simulator	154

<i>TABLE OF CONTENTS</i>	xi
8.3 Shadow Detection	156
A Material Wise Simulation Results	169
B Source Files List	182
C Source Codes	184
C.1 Getting Modtran Data	184
C.2 Convolving with sensor response	186
C.3 Sub-pixel / Full-pixel target planting in CameoSim Environment	188
C.4 CHIMES Codes	198
C.5 Adjacency Model Rendering	201

List of Figures

2.1	imaging system's capacity in terms of spatial and spectroscopic data fidelity.	11
2.2	Hyperspectral Scene as cube, a voxel is shown as reflectance. This reflectance is a composition of different materials such as Green Vegetation, Dry Vegetation, Soli and Kaolinite.	11
2.3	Electromagnetic Spectrum in Reflective and Emissive spectral regions. The spectrum is arranged from longer (Radio waves) up to shorter (Gamma waves). The visible spectrum is a smaller portion of the electromagnetic spectrum, therefore it is enlarged for elaboration	12
2.4	Components influencing Hyperspectral Remote Sensing.	13
2.5	Basic components for hyperspectral scene simulation [3]	14
2.6	Components of scene simulators.	15
2.7	Reflectivity, Absorptivity and Transmittivity of incident radiation.	22
2.8	Atmospheric absorption in the electromagnetic spectrum at 27 °C.[1]	26
2.9	Electromagnetic Propagation at material's boundary	27
2.10	Solid Angle	31
2.11	Bi-directional Reflection Distribution Function	32
2.12	Bi-directional Reflection/Transmission Distribution Function	33
2.13	Reflection Geometry illustrating parameters of Phong BRDF.	34
2.14	First row: BRDF model created by Phong equation having variable values for key paramters [1]. Second row: Measured BRDF of know material recorded in MERL benchmark dataset [2, 3].	35
2.15	DIRSIG big equation components [4, 5]. Yellow rays are emitted by the sun while orange rays are emitted from various objects in the environment. Ray "A" represent direct reflected radiance, "B" shows diffused reflected radiance, "C" is scattered into the sensor's LoS. Ray "D" is emitted by the target object. "E" is emitted from the environment and reflected from the target object. "F" is in-scattered from the atmosphere into sensor's LoS. "G" is background reflected. "H" is emitted by background and reflected off the target object and I is reflected and scattered into the sensor LoS.	36
2.16	ToA solar irradiance, clear-sky atmosphere, solar irradiance at target	37
2.17	solar downwelled scattered radiance.	38
2.18	Sky-view factor for a roof slope geometry	40

2.19	Solar radiance scattered towards the LoS of the sensor without coming in contact with the target. (path radiance)	41
2.20	Estimating $f(x)$ by $pdf(x)$: (a) bad pdf (b) uniform pdf (c) similar pdf	46
2.21	DIRSIG big equation components [4, 5]	52
2.22	OSSIM's radiative transfer equation components [4].	53
2.23	CameoSim data flow.	55
2.24	CameoSim-like recursive ray-tracing.	56
2.25	Adjacency components shown as dashed lines.	58
2.26	Adjacency effect in rugged terrains.[6, 7]	59
2.27	Ray tracing setup, at the diffuse and transparent object intersection the reflected rays (green) are spawned into new similar rays. In transparent objects transmitted rays (orange) pass through the object. When a shadow is cast due to occlusion towards a background region, shadow rays (red) are cast in the opposite direction of the shadow, towards the emitter. An eye ray (blue) is cast from the eye or sensor.	62
2.28	Path tracing.[8]	63
2.29	Hammersley sampling in a unit area	65
2.30	Poisson sampling in a unit area [9]	65
3.1	CHIMES Simulator, system level design overview. Sensor, Scene, BRDF, Renderer and Image Reconstruction modules are shown in different components as coloured boxes.	68
3.2	CHIMES components, data flow and user interaction methods	69
3.3	CHIMES Simulator simple GUI, showing a pre-rendering RGB view of Modesto Scene, partly showing sky and sun. Scene can be defined as python script or XML definitions file.	70
3.4	Reflectance cube retrieved by QUAC, K-means clustering is utilized for classifying the reflectance. Classified textures with 80, 120 K-means classes are shown in (a,b).	72
3.5	Terrain created from DEM and background skydome in CHIMES software (an RGB view).	73
3.6	Downwelled solar irradiance	74
3.7	Solar irradiance in it's first leg is incident at the data points on the hemispherical horizon of the scene. Data point are selected spherical coordinates where MODTRAN's scattering, transmission etc are computed.	75
3.8	Data points on the hemispherical coordinates that are scattering points for downwelled radiance. The blue face shows the plane where interpolation is required to construct the sky environment map.	75
3.9	Upwelled scattered radiance with spherical and altitude variability. Blue points show the geometric points where data is acquired from MODTRAN, at heights h_0, h_1, h_2	76
3.10	Degree of Polarisation with respect to zenith and azimuth	78

3.11	CHIMES sky and sun emitters.	78
3.12	CHIMES emitters.	79
3.13	Material ID texture of Selene scene.	80
3.14	Material ID texture of Modesto scene.	80
3.15	CHIMES Database structure	81
3.16	Selene texture data.	82
3.17	Flowchart illustrating recursive ray casting and estimating L_{sensor} during rendering loop.	84
3.18	Path tracing based volumetric scattering of monochromatic light rays in presence of aerosols. Scattering event generates a deviation, hence another branch of recursive ray.	85
3.19	Scene geometry is a 3D bounded region, the scene is partitioned into regions by generated random points or primitives. Labelled points are visual depiction of data structure nodes that are handled in kd-trees.	86
3.20	kd-tree of partitioned scene area shown in Figure 3.19.	86
3.21	(a) The data flow of the proposed TIAEM model which computes the adjacent upwelling radiance by evaluation of volumetric scattering with an spatial ROI. It depends on the difference of altitude between sensor and height of terrain at a given point on texture. (b) shows the evaluation procedure of averaging the background albedo of the ROI by the adjacency pixels (in red) for the test (target) pixel which is in orange/green.	91
3.22	Selene Airborne scene; altitude 0.944 Km . Scene simulated with upwelled scattered term for BOAEM and TIAEM.	92
3.23	Modesto Airborne scene; altitude 20 Km . Scene simulated with upwelled scattered term for BOAEM and TIAEM.	92
3.24	Flowchart of searching underlying atmospheric conditions parameters.	95
3.25	Atmospheric search based White and black panel estimates.	97
4.1	Components of multi-layered regression learning algorithm has three phases: (i) regression learning; (ii) feature learning; and (iii) classification. Inputs are radiance and reflectance cubes that are randomly sampled to find neutral parameter estimates for α_G and β_G . A kernel of size 3 used for kernel-based linear regression parameters are α_K and β_K , representing a more localized and homogeneous (either shadowed or non-shadowed regions) estimate. In Phase II, more discriminating parameters are found in the second layer of regression learning, which rectifies parameters estimated by the previous phase. Two non-linear filter layers are shown. Eventually, the classifier layer segregates shadowed and non-shadowed regions.	103
4.2	Regression learning reduces error $\hat{\epsilon}$ to find parameters α (total ground-reflected radiance) and β (path radiance).	104

4.3	White and black panel estimates from global search.	105
4.4	Intermediate Map extracted by proposed method on coarse (10×10) and fine (3×3) sliding window kernel. To highlight the shadow region $1/\tilde{t}$ is computed from Equation 4.17, whose values are shown in the bar. The magnitude above 1.0 is classified as shadowed region. As \tilde{t} is a ratio of same quantity therefore it has no units.	106
4.5	Kernel-based regression performs search for localized parameters α_K and β_K by pixels that are only within kernel.	107
4.6	Flowchart of the proposed multi-layered regression learning algorithm. Parameters α, β with subscript “ G ” denote global, and “ K ” denotes local (under the kernel). Moreover, the parameter subscript “ NS ” denotes non-shadowed and “ S ” denotes shadowed regions. ρ stands for reflectance, and t denotes threshold. In global search several pixels are randomly selected for regression learning and retrieval of TGRR. In local search a kernel is slid over the image where pixel observations are selected from the active kernel position.	109
4.7	Regression learning reduces error \hat{e} to find parameters α_S (total ground-reflected radiance in the shadow region) and β_S (path radiance in shadow region) and L_S (total radiance under shadow region).	110
4.8	Effect of filtering on a given pixel.	113
5.1	RGB color images of Selene and Modesto test scenes, respectively.	115
5.2	Commercially available fabric based calibration panels and tiles were planted in Selene scene [10]. Average reflectivity for calibration panels varies from 99.0% to 2.0%.	116
5.3	Location and reflectivity of calibration panels planted within the Ground Truth Selene scene [10].	117
6.1	White and black panel estimates from atmospheric search.	124
6.2	RGB color image for simulated and Ground Truth Selene scene. Atmospheric condition listed in Table 3.1.	126
6.3	ℓ_1 Norm error of simulated scenes with respect to Ground Truth in atmospheric condition of Table 3.1.	127
6.4	Mean radiance of ROIs shown in Figure 5.1.	128
6.5	Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GT’s.	130
6.6	RGB color image for simulated and Ground Truth Selene scene. Atmospheric condition is given in Table 3.1.	132
6.7	ℓ_1 Norm error of simulated scenes with respect to Ground Truth.	133
6.8	Mean radiance of ROIs shown in Figure 5.1	135
6.9	Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GTs.	136

6.10	ℓ_1 -norm error of simulated scenes with respect to Ground Truth in clear sky atmosphere. GT radiance is only used as reference for calculating error.	137
6.11	Radiance of white calibration panel in a (5×5) ROI. Simulated radiance are for clearsky while ground truth (GT) is under cloudy condition. GT is only used as a reference in this comparison.	138
6.12	RGB image from Simulation of Modesto scene with CHIMES, CameoSim with Ground Truth.	139
6.13	ℓ_1 Norm error of simulated scenes with respect to Ground Truth.	140
6.14	Mean radiance of ROIs shown in Figure 5.1	141
6.15	Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GT's	142
6.16	SELENE: Intermediate maps extracted by shadow detection methods before thresholding for a binary classification map.	145
6.17	SELENE: Binary classification map separating shadow and non-shadow regions.	146
6.18	MODESTO: Intermediate maps extracted by shadow detection methods before thresholding for a binary classification map.	148
6.19	MODESTO: Binary classification map (shadow and non-shadow regions.)	148
A.1	Flat Terrain: Radiance of a ROI (5×5) for grass near road. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.	171
A.2	Selene Flat Terrain: Radiance of a ROI (5×5) for Orange Perspex near grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.	172
A.3	Selene Flat Terrain: Radiance of a ROI (5×5) for White Panel. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.	173
A.4	Selene Flat Terrain: Radiance of a ROI (5×5) for Concrete. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.	174
A.5	Selene DEM Terrain: Radiance of a ROI (5×5) for Grass near road. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.	175

A.6 **Selene DEM Terrain:** Radiance of a ROI (5×5) for Orange Perspex near grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 176

A.7 **Selene DEM Terrain:** Radiance of a ROI (5×5) for White Panel. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 177

A.8 **Selene DEM Terrain:** Radiance of a ROI (5×5) for Concrete. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 178

A.9 **Modesto Flat Terrain:** Radiance of a ROI (5×5) for a mixture of soil and grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 179

A.10 **Modesto Flat Terrain:** Radiance of a ROI (5×5) for farm soil. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 180

A.11 **Modesto Flat Terrain:** Radiance of a ROI (5×5) for urban region of town. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated. 181

B.1 A list of library files developed in CHIMES along with the third party libraries. 183

List of Tables

2.1	Hyperspectral scene simulators, their developers, year of development, primary objective, spectral bands and model types. DISORT algorithm [11] is an integral part of MODTRAN which is used to calculate scattering and adjacency effect.	16
2.2	Radiometric quantities, symbols, their relationship and units [12].	35
2.3	Symbols in Equation 2.102 and their meaning, λ shows that quantities are function of wavelength.	52
2.4	Symbols in Equation 2.102 and their meaning, λ shows that quantities are function of wavelength.	54
2.5	Symbols in Equation (2.104-2.107) and their meaning.	57
2.6	Symbols in Equations (2.109) and (2.110) and their meaning.	60
3.1	MODTRAN parameters and their respective value for the Selene scene. Values in boldface are the variables that are searched for different values to fit the closest radiance. Other values are fixed for a specific scene according to the acquired scene's meta information and date/time.	94
6.1	NRMSE, ℓ_1 -norm error of radiance in Figure 6.4. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.	129
6.2	NRMSE, ℓ_1 -norm error of radiance in Figure 6.8. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.	135
6.3	NRMSE, ℓ_1 -norm error of radiance in Figure 6.8. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.	141

List of Abbreviations

AC	Atmospheric Compensation
ACE	Adaptive Cosine Estimator
AEM	Adjacency Effect Model
AGL	Absolute Ground Level
BOAEM	Background Only Adjacency Effect Model
BRDF	Bidirectional Reflection Distribution Function
BST	British Standard Time
BTDF	Bidirectional Transmission Distribution Function
CameoSim	CAMouflage Electro-Optic Simulator
CDF	Cumulative Distribution Function
COTS	Commercial-Off-The-Shelf
CHIMES	Cranfield Hyperspectral Image Modelling and Evaluation System
DEM	Digital Elevation Model
DoP	Degree of Polarisation
DN	Digital Number
DSM	Digital Surface Model
DSTL	Defence Science and Technology Laboratory
DIRSIG	Digital Imaging and Remote Sensing Image Generation
DOP	Degree Of Polarisation
DISORT	Discrete Ordinate Radiative Transfer
ELM	Empirical Line Method
EMR	Electromagnetic Radiations
EnMAP	Environmental Mapping and Analysis Program
FoV	Field Of View
GSD	Ground-Sampling Distance
GT	Ground Truth
HSI	Hyperspectral Image/Imaging
LIDAR	Light Detection and Ranging

LoS	Line of Sight
LULC	Land Use Land Cover
LUT	Look-up Table
LWIR	Long-Wavelength Infrared
MCSce	Monte-Carlo Scene
MIS	Multiple Importance Sampling
MODTRAN	MODerate resolution TRANsmission
MSI	Multispectral Image/Imaging
MWIR	Medium-Wavelength Infrared
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NIR	Near Infrared
NRMSE	Normalized Root Mean Square Error
NS	Non-Shadow
OSSIM	Optronics Scene Simulator
PICASSO	Parameterized Image Chain Analysis and Simulation Software
QUAC	QUick Atmospheric Correction
RGB	Red Green Blue
RGBN	Red Green Blue NIR
ROC	Receiver Operating Characteristics
ROI	Region Of Interest
RT	Radiative Transfer
S	Shadow
SAR	Synthetic Aperture Radar
SWIR	Short-Wave Infrared
TGRR	Total Ground Reflected Radiance
TIAEM	Texture Incorporated Adjacency Effect Model

Acknowledgements

I am extremely grateful to Dr. Peter Yuen for his support during this PhD study and related research. His dedication and engagement towards PhD research projects enable him to always spare ample time for discussion and to give prompt feedback. This attribute particularly proved very valuable during my studies and enabled me to improve my work continuously. I am thankful to him for his constructive criticism and valuable suggestions, which helped me shape this research more appropriately.

I also wish to acknowledge my sincere gratitude to CDS Doctoral Community lead Ms. Bea Kingdon, CDS researchers Dr Umair Soori and Dr Lounis Chemark for helping me gaining motivation throughout my studies. I am also greatly indebted to my fellow researchers Ayan Chatterjee and Murat Guneş who worked together with me on different research and development project.

I am also particularly thankful to Peter Godfree and Jonathan Piper of DSTL for their reviews during the course of this research and generally to DSTL for funding the project.

Finally, I am very grateful to my mother Najma Saba, my wife Dr. Qurrat Ul Ain and my family for their support without which I would not be able to pursue my PhD.

List of Publications

Journal

- **Usman A. Zahidi**; Peter W. T. Yuen, Jonathan Piper; Peter Godfree An End-to-End Hyperspectral Scene Simulator with Alternate Adjacency Effect Models and Its Comparison with CameoSim. Remote Sens. 2020, 12, 74. <https://doi.org/10.3390/rs12010074>.
- **Usman A. Zahidi**, Ayan Chatterjee, Peter W.T. Yuen, "A radiative transfer model-based multi-layered regression learning to estimate shadow map in hyperspectral images". MDPI Machine Learning and Knowledge Extraction, Volume 1, Issue 3, 2019, pp. 904-927. <https://www.mdpi.com/2504-4990/1/3/52>.
- Mengjia Ding; Peter W. T. Yuen; Jonathan Piper; Peter Godfree; Ayan Chatterjee; **Usman A. Zahidi**; Senthuran Selvagumar; David James; Mark Richardson "Design of a Tunable Snapshot Multispectral Imaging System through Ray Tracing Simulation." J. Imaging 2019, 5, 9. <https://doi.org/10.3390/jimaging5010009>.

Conference

- Umair Soori, Peter W. T. Yuen, **Usman A. Zahidi**, and David James "Chemical sensing of washing powder by multispectral imaging", AIP Conference Proceedings 2146, 020011 (2019); <https://doi.org/10.1063/1.5123698>.

Chapter 1

Introduction

1.1 Hyperspectral Imaging

Remote sensing technology has developed a diverse set of domain-specific acquisition instruments and scanning techniques. Hyperspectral Imaging (HSI) is one of them. Remote sensing enables retrieval of material properties by the acquisition of radiated or reflected electromagnetic radiations from the target of interest. The term Hyperspectral Imaging was first coined by [13] in 1985 for applications of imaging spectrometry. HSI has several challenges along in its design, development and deployment procedures, such as cost-effective sensor design that acquires realistic physical quantities such as radiance and reflectance from an observed scene. A major source of error is the estimation of atmospheric transmission and scattering effect. Selecting the appropriate scanning method is also a challenging pursuit.

There are several imaging spectrometry missions launched for the Earth-Observation and defence motivation. One of the early motivations for the development of HSI was the mineralogical mapping of surface soils and crops [13], a large application domain of HSI is still within the analysis and observation of vegetation which accounts for up to 70% of the application [13]. Military and defence applications are also one of early interest of HSI. The beginnings of imaging spectrometry of the Earth are rooted in the launch of Landsat-1, at the time called ERTS-1, in 1972. A series of Landsat satellites are launched since then and the latest ones are Landsat-7 and Landsat-8 multispectral satellites. AVIRIS development was begun in 1984 and the imager first flew aboard a NASA ER-2 aircraft at 20 km altitude in 1987. Since then it has gone through major upgrades as technology changed in detectors, electronics and computing. The European PROBA/CHRIS mission launched in 2001 provides 18 spectral channels covering the VNIR region [14]. WorldView-3 (WV 3) is a commercial Earth observation satellite

owned by DigitalGlobe. It was launched on 13 August 2014 to become DigitalGlobe's sixth satellite in orbit, joining Ikonos which was launched in 1999, QuickBird in 2001, WorldView-1 in 2007, GeoEye-1 in 2008, and WorldView-2 in 2009. WorldView-3 provides commercially available panchromatic imagery of 0.31 m (12 in) resolution, eight-band multispectral imagery with 1.24 m (4 ft 1 in) resolution, shortwave infrared imagery at 3.7 m (12 ft 2 in) resolution, and CAVIS (Clouds, Aerosols, Vapors, Ice, and Snow) data at 30 m (98 ft) resolution.[15]. The Environmental Mapping and Analysis Program (EnMAP) is a German hyperspectral satellite mission that aims at monitoring and characterising the Earth's environment on a global scale. EnMAP serves to measure and model key dynamic processes of the Earth's ecosystems by extracting geochemical, biochemical and biophysical parameters, which provide information on the status and evolution of various terrestrial and aquatic ecosystems [16].

Hyperspectral imagers typically designed as a scanning or snapshot integration time devices. Snapshot HSI imagers are a recent development in the industry and some products have become commercially available [17]. This technology has enabled HSI applications in a wide variety of domains. Despite being relatively expensive technology particularly in the SWIR/LWIR spectrum, HSI is still actively applied technology with applications in fault tolerance, life-sciences, food processing, quality assurance, aerial archaeology and monitoring is a vast variety of applications.

1.2 Scene Simulation

Hyperspectral imaging sensors require different materials for detection of particular wavelength radiation. The VNIR range detectors are typically constructed from Silicon which is cheaper and relatively easier to fabricate. Detectors for the range of SWIR/MWIR and LWIR are for example InGaAs and Ge therefore the cost of sensors are drastically higher. Additionally the cost of planning, pre and post processing and execution of data acquisition from airborne and satellite based platforms is a time and cost consuming process. Therefore correct prediction of sensor performance during operation can prove vital during the design phase. In this type of simulator emphasis is particularly on the sensor and atmospheric modelling. When a simulated scene is generated for performance analysis of data processing algorithms/frameworks. A more physically accurate simulation is required that incorporates artefacts due to radiometric, geometric, atmospheric and sensor effects to name a few. Typically rendering based solutions are used to develop such

simulators.

One application of performance analysis of data processing algorithms is for realistic assessment of detectability and vulnerability of targets is important in both civil and military applications. This requires the acquisition of high fidelity hyperspectral remote sensing images (HSI) for both targets and the background materials of the scene, which is time and cost-ineffective and, in some cases, it is even not possible to achieve through experimental trials. This drives the motivation of research to simulate remote sensing HSI data to allow the assessment of target detection capability under a variety of scene backgrounds, atmospheric conditions, sensor characteristics, ambient temperatures and light sources. Despite the existence of several commercial-off-the-shelf (COTS) HSI simulators within the remote sensing community, procurement, modification or enhancement of existing COTS simulation packages are prohibitive due to the export control and copyright protection issues. The objective of the present work is to establish a HSI simulator known as the Cranfield Hyperspectral Image Modelling and Evaluation System (CHIMES), which features an automatic atmospheric parameterization through a search algorithm to match the ground radiance of a given scene, and the proposal of two adjacency models to enhance the quality of the scene simulation. The output of the CHIMES is validated by a ground truth HSI scene and also compared with a COTS package CAMouflage Electro-Optic Simulator (CameoSim) [18].

Several Hyperspectral Scene simulators with customized focus have been developed by several research and commercial organisations. One of the earliest attempts such as Forecasting and Analysis of Spectroradiometric System Performance (FASSP) [19] offered statistical modelling of a scene primarily for sensor performance assessment. It did not offer ray tracing and advanced atmospheric processing in the scene. In its inception, CameoSim was only focussed at Camouflage evaluation, which later evolved into a comprehensive Hyperspectral simulator. However, till date, CameoSim does not perform volumetric scattering in the renderer which may limit the capability of realistic atmospheric modelling of the aerosol interactions with light. Optronic Scene Simulator (OSSIM) [4] is another simulator which is primarily a thermal modelling tool integrates external ray-tracer for computation. Rochester Institute of Technology's DIRSIG 4 [20] is another comprehensive simulator with image generation capability in Visible to Near Infrared (VNIR) to Long Wave Infrared (LWIR) spectral range. It also performs polarimetric, Synthetic Aperture Radar (SAR), Light Detection and Ranging (LIDAR) simulations. However, DIRSIG4 does not possess its component of adjacency modelling, which

is later implemented in DIRSIG5 through volumetric path tracing in a localised fashion. ATCOR [21] is an atmospheric compensation algorithm which also implements a localized adjacency model which is based on a physical model but does not perform rendering based volumetric scattering.

1.3 Motivation of Research

We learned from section 1.2 that although there are several alternate HSI scene simulators available in the market, however, there still exist many important unfilled gaps such as having a solution that offers a rendering based HSI simulation system which performs volumetric path-tracing and has an adaptive localised physics-based adjacency effect model which is integrated into the renderer. Therefore, we developed the CHIMES simulator from scratch, firstly to gain comprehensive understanding and control on HSI modelling methodologies and secondly to implement a localised physics-based adjacency model.

The motivation behind Hyperspectral scene simulation is driven by following major factors, that are;

- Time and Cost Efficient Reconfiguration, in terms of;
 - Scene
 - Atmosphere
 - Sensor
 - Light sources (active or Passive)
 - Temperature map

These factors are actually components of scene simulator, some are mandatory such as scene, atmosphere, sensor and light sources, therefore these are essential components of CHIMES simulator. However thermal modelling (temperature map) is an optional component which is not an inherent part of many simulators including CameoSim, which incorporates Taitherm for thermal modelling. This feature is not included in CHIMES simulator.

1.3.1 Sponsor's Motivation

Most COTS HSI simulators are developed in customized fashion which is typically intended for a limited user-base, either associated with a government institute or only to Defence industry. Therefore, these simulators are export controlled and their source-codes are strictly inaccessible.

- A major motivation in the development of this project is to gain detailed design level understanding of an HSI simulator in a rendering based system.
- During this project the Cranfield University has gained access to a code-base which could be further enhanced and modified.

1.4 Problem Statment

- To construct a software environment that takes the reflectance image of any given spectral resolution and creates a radiance image by incorporating path-tracing, volumetric scattering, sensor convolution.
- To construct a dynamic and localized physics-based adjacency effect model integrated into a rendering system.
- To develop a methodology for estimating an underlying atmospheric parameters when some inscene reflectance materials are known.
- To develop a Radiative Transfer model based shadow detection method.

1.5 Scope and New aspect of work

1.5.1 Scope

This project is planned into two phases, during phase one the goal is develop the essential components of a scene simulator and integrate them together to be able to perform simulations. In the second phase the focus is on the research and development of the localised adjacency effect modelling, volumetric scattering and automated searching of MODTRAN atmospheric parameters.

Due to lack of resources such as time and number of people working on the development of this research project. There are certain limitation defined during the planning phase. This include simulation within Reflective spectral range i.e. (0.4 – 2.5) i.e thermal range is not supported. The scattering model only takes elastic scattering into account such as Rayleigh and Mie scattering only, ignoring Raman Scattering. The illumination source is only sun and sensor orientation is nadir only. Illumination data is calculated only from MODTRAN.

1.5.2 New Simulator and Features

During this research, we developed a new rendering-based end to end Hyperspectral scene simulator CHIMES (Cranfield Hyperspectral Image Modelling and Evaluation System) from scratch, which generates nadir images of passively illuminated 3-D outdoor scenes in Visible, Near Infrared (NIR) and Short-Wave Infrared (SWIR) regions.

1.5.3 Alternate Adjacency Modelling

Adjacency effect is the influence of the most abundant background material within a scene on a smaller target due to upwelled scattering. As it depends on the degree of scattering coefficient and height of the sensor it may vary the at-sensor radiance between 10-20% compared to ground measurements.[22] Most existing COTS HSI simulators employ DISORT for modelling adjacency effect, which uses average reflectance of the whole scene to estimate adjacency effect on target material. However, Richter et. al. proposed higher spatial variation in their adjacency effect model by introducing a limited region of effective adjacency influence, which is incorporated into ATCOR [21]. The effective region in their work is defined based on the volume of medium between target and sensor. ATCOR performs adjacency calculation based on a fixed height of terrain for every pixel, it does not performs volumetric scattering and estimates contrast reduction by supplementing a term in existing adjacency models. CHIMES incorporated their adjacency model in the renderer, which evaluates upwelled scattered radiance based on varying heights across the terrain. Furthermore, the contrast term is replaced by volumetric scattering along the rays. This setup creates a unique combination which is not reported in the public domain.

1.5.4 Search Method for approximating underlying atmospheric conditions

The COTS HSI simulators typically require user input for generating atmospheric data. CHIMES introduce a feature to facilitate searching of MODTRAN's parameters that are listed in Table 3.1, therefore it is a characteristic feature which is integrated into CHIMES simulator. The presence of aerosols or water droplets in the atmosphere is a complicated phenomenon to model as their properties, such as their sizes, densities and their distributions are subjected to many atmospheric factors which makes them difficult to predict. We provide an automated estimation method to guess the underlying atmospheric condition MODTRAN is run with varying aerosol optical thickness and its total ground reflected radiance (TGRR) is compared with TGRR of known in-scene material. The Goodness of fit is evaluated in each iteration, and MODTRAN's output with the best fit is selected. Several candidate solutions may produce the best fit, one among the candidate solutions is picked in our experiments.

1.5.5 Radiative Transfer Model-based Shadow Detection

- Presence of shadows causes significant challenges for both satellite and airborne data analyses. Shadows are typically not compensated by Atmospheric compensation modules and therefore retrieving reflectance input where shadows are not accounted for, causes higher error in simulated images. It is therefore important to compensate shadows for improving the quality of simulation. We provide a multi-layered regression learning-based recovery of radiance components, i.e., total ground-reflected radiance and path radiance from QUAC reflectance and radiance images of the scene. These decomposed components represent terms in the RT equation and enable us to relate variable illumination in a scene mathematically. The retrieved components are used to create an intermediate map which is thresholded to construct a shadow map. This approach towards shadow detection is not reported in literature to our best knowledge.

1.6 Thesis Organisation

The thesis is as follows; Chapter 1 (this chapter) outlines the motivation and the objective of the present work; Chapter 2 establishes the fundamental background literature for hyperspectral scene simulation and radiative transfer Chapter 3 constitutes the main topic of the thesis which outlines the skeleton of the proposed CHIMES simulator with more details on the proposed adjacency models, ray tracing and radiometric integrators. Chapter 4 provides theoretical and implementation details about proposed shadow detection algorithm. Chapter 5 gives the background of experimental data and details the acquisition methods. Chapter 6 provide results for all datasets of simulation and shadow detection. Chapter 7 comprises of the discussion on the results while Chapter 8 summarizes the main achievements of the present work and present the potential future work. appendices gives detailed simulation results achieved by the proposed CHIMES and CameoSim, provide list of developed software library files and also lists some source-codes.

Chapter 2

Fundamentals of Hyperspectral Scene Simulation and Radiative Transfer

Remote sensing is a natural extension of the human need to explore and understand its environment without coming in physical contact with the target. Hyperspectral remote sensing is one of the powerful tools in the field of remote sensing [23]. It is a blend of the spectroscopic and spatial information about a scene having high spectroscopic data fidelity together with spatial one. The spectral resolution is in the magnitude of 10-20 nm, therefore data is acquired from 100 to 500 or more bands. Figure 2.1 gives an overview of where hyperspectral remote sensing stands compared to other imaging and sensing systems in terms of spectroscopic and spatial data fidelity.

Data captured from a Hyperspectral sensor is stored as a cube where two dimensions represent spatial data and the third represents the spectra. The unit information in a hyperspectral image is a volume pixel which is also referred to as Voxels. Figure 2.2 shows a hyperspectral data cube, reflectance graph of a voxel and potential components of a reflectance observation. It shows that the reflectance is composed of different types of vegetation, soil and Kaolinite components [24]. Spectroscopy enables us to observe characteristics of materials by their unique interaction pattern with electromagnetic radiations (EMR), which is due to their distinctive molecular formation. The exposure of EMR on material causes the transition of atoms, molecules or ions from one energy level to another, the reflected light thus represent this particular absorption or emission when radiated back towards the sensor. The sensor then decomposes the EM radiation by dispersion and represents the amount of per wavelength energy in terms of Digital Number (DN), which is a raw discrete representation of a particular wavelength.

Figure 2.2 shows the scene within $0.4 \mu\text{m}$ to $2.5 \mu\text{m}$ range. The number of bands in a given range forms the width of the band in terms of wavelength; this is referred

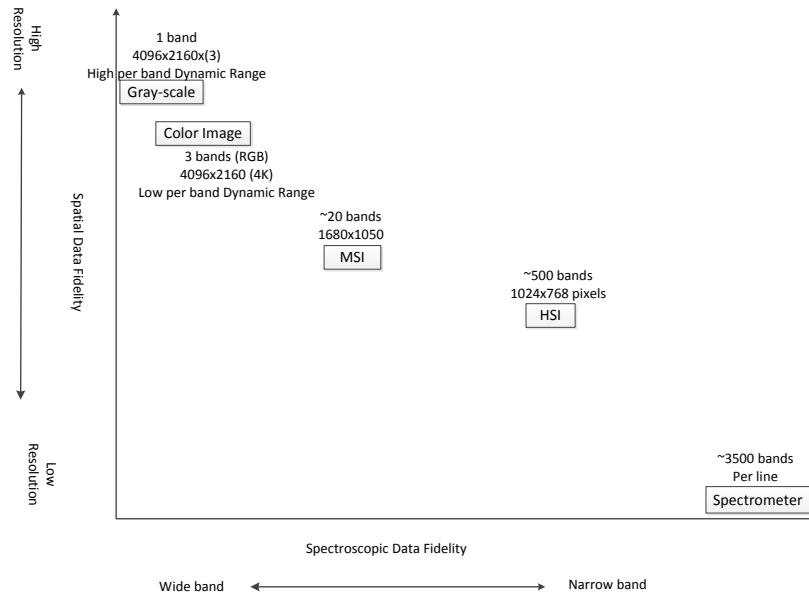


Figure 2.1: imaging system’s capacity in terms of spatial and spectroscopic data fidelity.

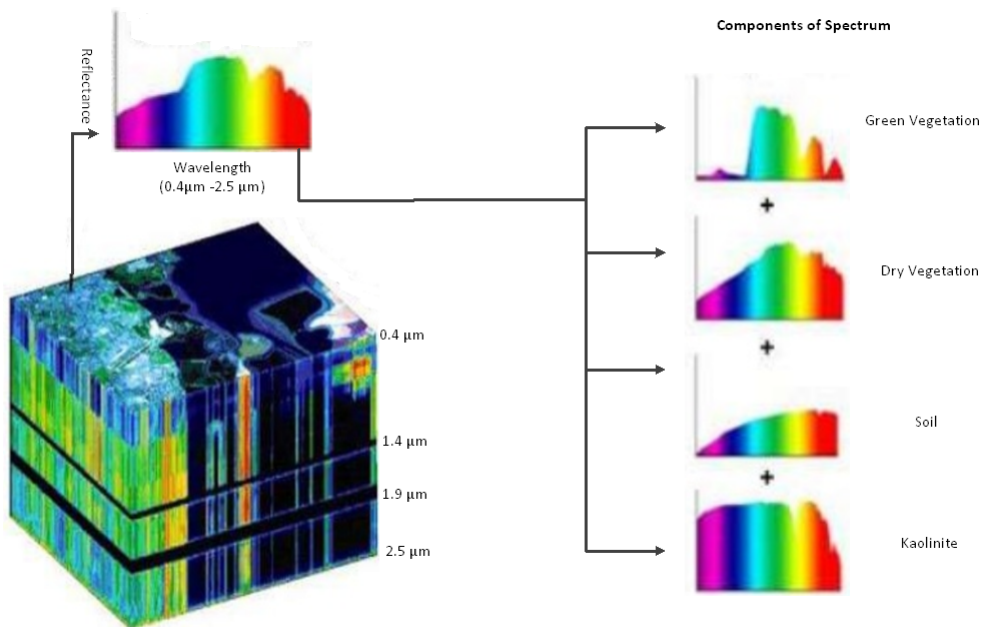


Figure 2.2: Hyperspectral Scene as cube, a voxel is shown as reflectance. This reflectance is a composition of different materials such as Green Vegetation, Dry Vegetation, Soli and Kaolinite.

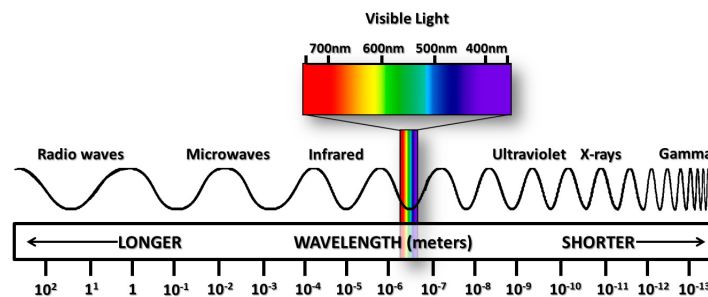


Figure 2.3: Electromagnetic Spectrum in Reflective and Emissive spectral regions. The spectrum is arranged from longer (Radio waves) up to shorter (Gamma waves). The visible spectrum is a smaller portion of the electromagnetic spectrum, therefore it is enlarged for elaboration

to as spectral resolution. Figure 2.3 shows the electromagnetic spectrum segregated in reflective and emission bands. A remote sensing observer can be mounted on an airborne vehicle or mounted on the satellite when capturing a natural scene, therefore it summates divergence due to atmosphere in material's reflectance together with its gains and effects. This observed measurement is called radiance.

During the day time, due to direct solar illumination, the sensor can measure the radiance in the visible ($0.4 \mu\text{m} - 0.7 \mu\text{m}$), Near Infra-red ($0.7 - 1.1 \mu\text{m}$) and SWIR (Short Wavelength Infrared, $1.1 - 3.0 \mu\text{m}$) regions. However during the night due to the lack of solar illumination, remote sensing is not possible in reflective region, therefore emission spectra are used instead, such as MWIR (Medium Wavelength Infrared, $3.0 - 5.5 \mu\text{m}$), LWIR (Long-Wavelength Infrared, $7.7 - 14 \mu\text{m}$). The physical basis of image content in VNIR is a combination of broadband reflectance differences between objects (shape) and illumination variation due to the 3D nature of the scene (shading). In MWIR and LWIR variation in object temperature and emissivity is the source of image structure [12].

2.1 Components of Hyperspectral Remote Sensing

Hyperspectral remote sensing requires understanding its four major components such as Material Spectroscopy, Radiative Transfer, Imaging and Sensor Systems and Data processing, as shown in figure 2.4. Material spectroscopy accounts for reflective/radiative behaviour of material on interaction with EMR. Radiative transfer is the physical phenomenon of energy transfer in the form of electromagnetic radiation. The propagation of

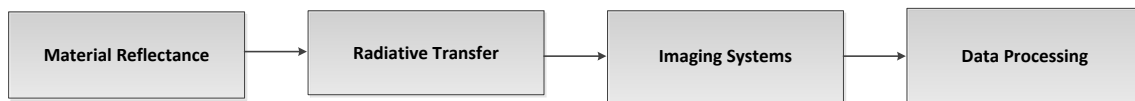


Figure 2.4: Components influencing Hyperspectral Remote Sensing.

radiation through a medium is affected by absorption, emission, and scattering processes. The design and fabrication and evaluation of Imaging Systems such that it may capture both spatial and spectral data simultaneously and its constituent materials do not affect the observation is also an important aspect of Hyperspectral remote sensing. Finally, data processing includes methods and algorithms that extract useful information from a mixture of signal and noise.

2.2 Hyperspectral Scene Simulation

Hyperspectral scene simulation is a type of inverse problem in which a hyperspectral radiance is to be reconstructed ideally from RGB or multispectral reflectance image. The most adopted approach to scene simulation is due to [19], which is shown in figure 2.5.

The first challenge in scene simulation is reconstruction of hyperspectral reflectance from mutispectral input. The second one is reflectance to at-sensor (observed) radiance conversion based on the radiative transfer model such that the scene statistical characteristics are recovered in the hyperspectral domain within the physics-based constraints. Sensor observation adds sensor model as a vital constituent of the reconstruction process. Atmospheric effects such as transmission, scattering and adjacency effects are typically added through a radiative transfer code such as MODTRAN, LibradTran, FASSCODE etc. The diversity of application areas for HSI scene modelling drives different motivations behind the development of such models and software applications. A detailed list of existing simulators, their motivation and capacity is covered in subsequent sections with a brief overview in Table 2.1.

2.2.1 Existing Simulators

In this section we provide details about existing HSI simulators. During this research results of CHIMES simulators are published in [18] but the details are not covered in this

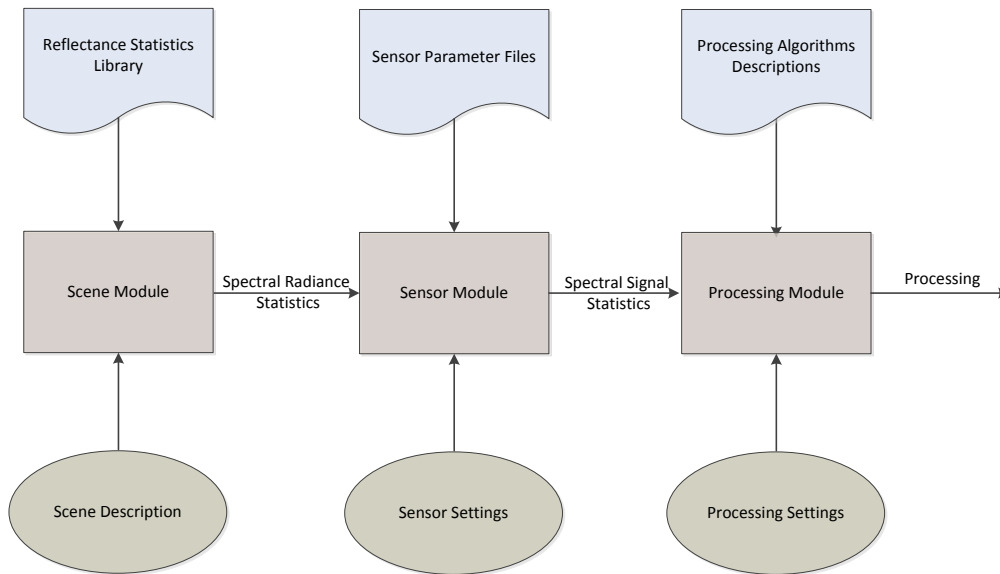


Figure 2.5: Basic components for hyperspectral scene simulation [3]

section. However it is covered in detail in chapter 3.

2.2.1.1 Forecasting and Analysis of Spectroradiometric System Performance (FASSP)

FASSP is one of the earliest attempts in this area, which pursued a statistical parametric modelling approach. FASSP is a statistical model for the analysis of earth observing data acquired by optical spectral imaging systems. The predominant surface parameters considered are the spectral reflectance mean vector and spectral covariance matrix [25].

FASSP architecture provides identification of the primary components of the scene for the simulation [19] which is shown in Figure 2.5, these components are one of the building blocks for our proposed simulator, however, we extended their components to include rendering and image reconstruction as supplementary ones. Our revised components of a scene simulator and data flow is shown in Figure 2.6 [18].

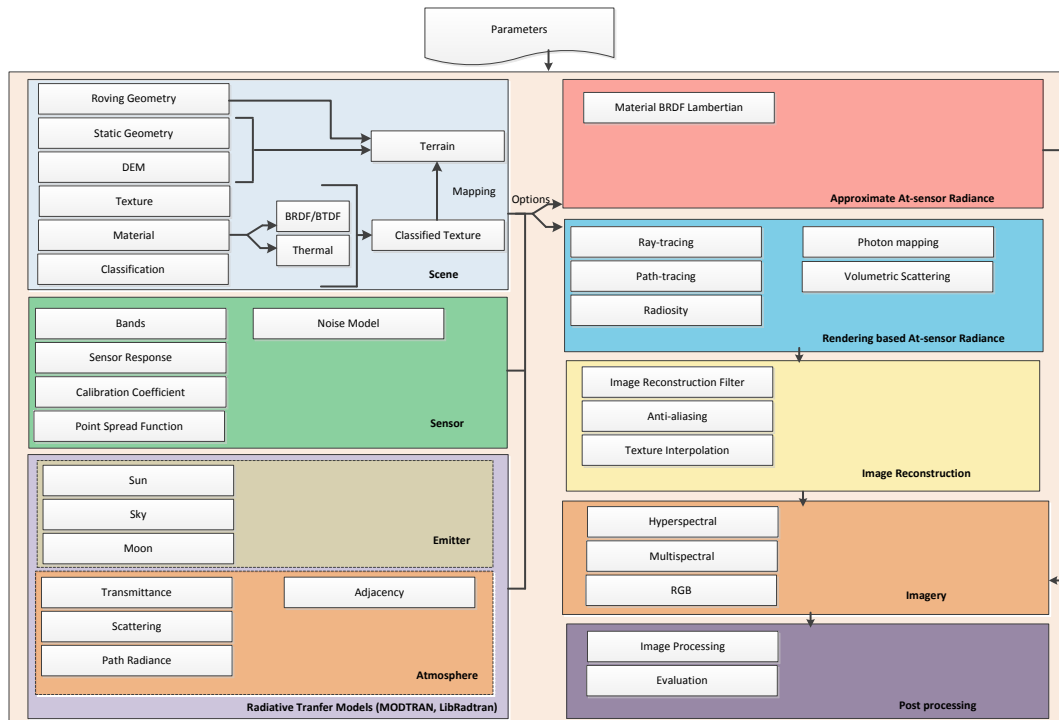


Figure 2.6: Components of scene simulators.

2.2.1.2 Digital Imaging and Remote Sensing Image Generation (DIRSIG)

DIRSIG adopts an image-chain approach to Remote-Sensing simulation. It is capable to simulate passive broadband, multi-spectral, hyper-spectral, low-light, polarized, active laser radar, and synthetic aperture radar datasets. Its components include bi-directional reflectance distribution function (BRDF) predictions of a surface, time and material-dependent surface temperature predictions, to the dynamic viewing geometry of scanning imaging instruments on the agile ground, airborne and space-based platforms [20, 26, 27, 28]. In a recent update called DIRSIG5 [29] Metropolis Light Transport (MLT) based path tracing has been used. Atmospheric processing can be performed by both MODTRAN5 and MODTRAN6 [30][18].

Table 2.1: Hyperspectral scene simulators, their developers, year of development, primary objective, spectral bands, primary objective, spectral bands and model types. DISORT algorithm [11] is an integral part of MODTRAN which is used to calculate scattering and adjacency effect.

Simulators	Developed By	Primary Objective	Spectral Bands (μm)	Model Type	Adjacency-Effect	Noise Model
FASSP (1996-)	MIT Lincoln's Lab	Sensor Modelling	0.3 - 2.5	MODTRAN-based Statistical Estimation	DISORT	Yes
CameoSim (1998-)	Lockheed Martin/Insys	Camouflage Assessment	0.3 - 20.0	Light Transport (Radiosity)	DISORT	No
DIRSIG 1-4 (1997-2015)	RIT	HS Image Generation	0.3 - 20.0	Light Transport (Whitted Ray-tracing)	DISORT	Yes
DIRSIG 5 (2015-)	RIT	HS Image Generation	0.3 - 20.0	Light Transport (Metropolis Light Transport)	MODTRAN6	Yes
OSSIM (2004-)	Denel SOC Ltd	Countermeasure	0.4 - 20.0	Light Transport (Safir-based Ray-tracing)	DISORT	Yes
MCSce (1999-)	Spectral Science	Sensor Modelling	0.3 - 20.0	Statistical	DISORT	N/A
ATCOR (2006-)	ReSe	Atmospheric Compensation	0.3 - 2.5	MODTRAN-based Radiance Estimation	Own Model	Yes
PICASSO (1999-)	MIT Lincoln's Lab	Sensor Modelling	0.3 - 14.0	MODTRAN-based Radiance Estimation	DISORT	Yes
EnMap (2009-)	DLR	Sensor Modelling	0.3 - 2.5	MODTRAN-based Radiance Estimation	DISORT	N/A
CHIMES (2017-)	Cranfield University	HS Image Generation	0.3 - 2.5	Light Transport (Path-tracing)	DISORT	No

2.2.1.3 CAMouflage Electro-Optic Simulator (CameoSim)

CameoSim is a commercial scene simulator originally developed for the assessment of camouflage systems, from the far Infra-Red, visible and to the UV with HSI capabilities [31]. It uses raytracing with a Monte-Carlo radiosity light-transport scheme. All the geometric objects forming the synthetic environment are modelled using textured faceted structures. Texel values in these textures are mapped to the real materials associated with them [32, 33]. These materials can have measured properties using one of several BRDF models, with optional transparency. It supports moving observers and players, with moving parts. It can handle large complex scenes with billions of pixels and is applicable to ground, ocean, air and space scenarios. It can model plumes and flares. It predicts surface temperatures, with optional thermal shadowing, using weather measurements. Atmospherics can be generated using MODTRAN 5, or measured atmospherics imported. Environmental maps based on High Dynamic Range (HDR) images can be imported for scene simulation [18].

2.2.1.4 Optronic Scene Simulator (OSSIM)

OSSIM is another HSI scene simulator that creates synthetic images of arbitrary complex scenes by rendering, the spectral range is in the visual and infrared (IR) bands, covering the 0.4–20 μm spectral region. These images are radiometrically accurate and based on theoretical physics models. To allow for the subtleties and full scope of variability in atmospheric attenuation, the simulation employs all capabilities of the MODTRAN. The geometrical shape of objects and the terrain topography is described in terms of a three-dimensional complex hull, consisting of a set of flat, convex facets or polygons. Each polygon is assigned spectral radiometric properties, temporally variable spatial texture properties and radiometric properties. Polygons can also be partially transparent to represent gas clouds, countermeasure flares or aircraft plumes [4, 34, 1, 35] [18].

2.2.1.5 Monte-Carlo Scene (MCScene)

MCScene is also a rendering-based HSI scene simulator. It incorporates all optical effects important for solar-illuminated and thermal scenes, including molecular and aerosol scattering, absorption and emission and surface scattering with material-dependent bidirectional reflectance distribution functions (BRDFs), multiple scattering events, surface adjacency effects, and scattering, emission and shading by clouds, for arbitrary solar il-

lumination and sensor viewing geometries. The “world” of the simulation is a cube that encloses a user-definable atmosphere containing molecular species, aerosols, and clouds, and a terrain representing the ground. The sensor spatial and spectral resolution, its location, and the viewing angle are also specified. 3D objects can also be inserted into the scene. A particular strength of MCScene is that a simulation can be data-driven. Terrain information can be imported from United States Geological Surveys (USGS) digital elevation maps. Surface reflectance or emissivity/temperature maps can be derived from collected imagery, thus incorporating natural spectral and spatial texturing into a simulation [36, 37, 38, 39, 18].

2.2.1.6 Parameterized Image Chain Analysis & Simulation Software (PICASSO)

PICASSO is also one of the end-to-end HSI image simulation or image chain analysis tools which estimate at-sensor radiance based on MODTRAN. Simulation flow begins with a description of the remote sensing system to be modelled, in terms of standard engineering parameters (e.g., primary aperture diameter, effective focal length, focal plane array detector size, focal plane array operating temperature, etc.). PICASSO also requires an input earth scene that can serve as ground-truth. The principal output of PICASSO is a simulated image, having the characteristics that would be produced by the remote sensing system under simulation. In addition to this simulated image, PICASSO produces figures of merit commonly used throughout the remote sensing industry to characterize image quality. These can include such metrics as SNR for given radiance; plots of the system transfer function (STF) and point-spread function (PSF) or overall image quality as measured by the National Imagery Interpretability Rating Scale (NIIRS) [40, 41, 18].

2.2.1.7 The Environmental Mapping and Analysis Program (EnMAP)

ENMAP is a German hyperspectral satellite mission that aims at monitoring and characterising the Earth’s environment on a global scale. The hyperspectral imager is a push-broom type consisting of two prism imaging spectrometers—one for VNIR (Hypex VNIR1600) and one for SWIR (Hypex SWIR320m). As a precursor of the mission, EnMAP simulator was developed. The EnMAP scene simulator can automatically generate realistic EnMAP-like data under a set of user-driven instrument and scene parameters. Radiance and digital numbers data are generated by five sequential processing modules which can produce data over a range of natural environments, acquisition and illumination

geometries, cloud covers, and instrument configurations. The latter include the simulation of data nonuniformity in the spatial and spectral domains, spatially coherent and noncoherent instrumental noise, and instrument's modulation transfer function (MTF). Realistic surface patterns for the simulated data are provided by existing remote-sensing data in different environments, from dry geological sites to green vegetation areas. A flexible radiative transfer simulation scheme enables the generation of different illumination, observation, and atmospheric conditions [16, 18].

2.3 Drawbacks in existing simulators

Most existing and newly built simulators are based on MODTRAN Radiative Transfer codes, where MODTRAN is executed in several iterations to encompass whole scene geometry. In Table 2.1, we particularly classify simulators radiance output into rendering-based or statistical/MODTRAN based estimation. The effect of DEM in the case of rugged terrain is captured by rendering-based simulators only, therefore those performing rendering techniques result in an accurate estimation of radiance. CameoSim however, does not perform phase function based scattering during the rendering process, therefore it may not model the volumetric scattering effect in the scene. DIRSIG and MCScene provide volumetric scattering feature in their simulators, however, their adjacency model is based on MODTRAN's DISORT algorithm which unfortunately supports one material for estimating the adjacency material for the whole scene. [22] shows that their test scene shows one adjacency effect near water-front and other near vegetation within a scene. Therefore it is important to have a localised adjacency model support in an HSI simulator. Only DIRSIG5 has the support for localised adjacency effect employing ray tracing, which ignores the dependency on the sensor height and consequently the volume of air between terrain and the sensor. Moreover, to the best of our knowledge, all COTS simulators require the user to input the parameters of RT code (MODTRAN) to generate atmospheric data, which requires trial and error search for the end-users.

2.4 Radiative Transfer

Radiative transfer is the physical phenomenon of energy transfer in the form of electromagnetic radiation. The propagation of radiation through a medium is affected by absorption, emission, and scattering processes. Radiative transfer has application in a wide

variety of subjects including optics, astrophysics, atmospheric science, and remote sensing [42]. We will investigate three important aspects of radiative transfer in this chapter that are; Propagation and interactions of EM radiations, Radiometry and Atmospheric effects.

2.5 Light Material Interaction

2.5.1 Beer-Lambert Law

The Beer-Lambert law or simply Beer's law is the linear relationship between absorbance and transmittance of electromagnetic radiation. The general Beer-Lambert law is usually written as equation (2.1).

$$A = \mu_{\lambda} R c \quad (2.1)$$

where A is the measured absorbance, μ_{λ} is a wavelength-dependent absorptivity coefficient also called molar absorptivity, R is the path length, and c is the chemical compound concentration.

$$A = \sum_{i=1}^n \varepsilon_i R c_i \quad (2.2)$$

where ε_i is the wavelength-dependent molar absorptivity coefficient with units of $M^{-1}cm^{-1}$. The λ subscript is often dropped with the understanding that a value for ε is for a specific wavelength. If multiple species that absorb light at a given wavelength are present in a sample, the total absorbance at that wavelength is the sum due to all absorbers given by equation 2.2, where n denotes number of absorber.

Transmittance τ is given as the ratio between incident and emitted power, as shown in Equation (2.3), where I is the intensity of light after it passes through the sample and I_o is the initial light intensity.

$$\tau = \frac{I}{I_o} \quad (2.3)$$

Equation 2.4 gives the relation between A and τ , which is found in [43].

$$A = -\log(\tau) = -\log\left(\frac{I}{I_o}\right) \quad (2.4)$$

2.5.2 Limitations of the Beer-Lambert law

The linearity of the Beer-Lambert law is limited by chemical and instrumental factors. Causes of non-linearity include:

- Deviations in absorptivity coefficients at high concentrations ($> 0.01M$) due to electrostatic interactions between molecules in close proximity.
- Scattering of light due to particulates in the sample fluorescence or phosphorescence of the sample.
- Changes in refractive index at high compound's concentration shifts in chemical equilibrium as a function of concentration.
- Non-monochromatic radiation, deviations can be minimized by using a relatively flat part of the absorption spectrum such as the maximum of an absorption band.

2.5.3 Extinction of Radiation

A beam of radiation is attenuated by absorption and scattering. This reduction of intensity is called extinction. The extinction coefficient is related to absorption and scattering coefficients.

$$\mu_e = \mu_a + \mu_s \quad (2.5)$$

where μ_e is coefficient of extinction, μ_a is coefficient of absorptivity and μ_s is coefficient of scattering. All these coefficients have unit m^{-1} . The Beer-Lambert law is also applied to describe the attenuation of solar radiation as it travels through the atmosphere. In this case, there is the scattering of radiation as well as absorption. More details about different forms of Beer-Lambert law and relevant derivation is provided in [43]. Transmittance of radiation through the atmosphere, and extinction coefficients are related by Equation (2.6), where τ is transmittance and R is path length of radiation in the medium.

$$\tau = \exp^{-\mu_e R} \quad (2.6)$$

2.5.3.1 Absorptivity (α), Reflectivity (ρ) and Transmissivity (τ)

If the amounts of radiation energy absorbed, reflected, and transmitted when radiation strikes a surface are measured in percentage of the total energy in the incident electro-

magnetic waves. The total energy would be divided into three groups; they are called absorptivity (α), reflectivity (ρ) and transmissivity (τ) [1], as shown in Figure 2.7.

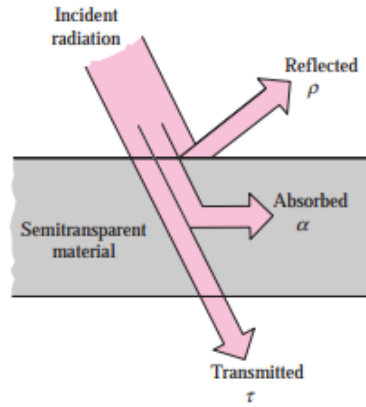


Figure 2.7: Reflectivity, Absorptivity and Transmittivity of incident radiation.

$$\alpha + \rho + \tau = 1 \quad (2.7)$$

In case of absorptive material, the fraction of irradiation absorbed by a surface is termed as absorptivity. Reflectivity is the fraction reflected by the surface. Transmissivity is the fraction transmitted by the surface.

2.5.3.2 Emissivity

Emissivity ε is the ratio of radiant emittance of a real and black body at temperature T . It is given by Equation (2.8).

$$\varepsilon = M_e / M_e^o \quad (2.8)$$

where M_e is the radiant emittance of real body M_e^o is the radiant emittance of a black body at the same temperature as that of the real body. The spectral emissivity is given by Equation (2.9).

$$\varepsilon_\lambda = M_{e,\lambda} / M_{e,\lambda}^o \quad (2.9)$$

Emissivity and reflectivity are related to each other by Equation (2.10).

$$\varepsilon = 1 - \rho \quad (2.10)$$

2.5.4 Rayleigh scattering

Lord Rayleigh explained why sky looks blue during day time and changes its brightness from dawn to dusk. He showed that scattering by air molecules was sufficient to produce the observed effect. Rayleigh scattering has following characteristics;[44]

- The amount of scattered light varies nearly as the inverse fourth power of wavelength, hence the clear sky is predominantly blue and not colour of sunlight.
- Spatial distribution of scattered light bears a simple relationship to the direction of observation Equal amount of light are scattered into the forward and backward hemispheres.
- Light scattered at 90° is almost completely polarized.

The size of a scattering particle is often parameterized by the ratio as;

$$x = \frac{2\pi r}{\lambda} \quad (2.11)$$

where r is its characteristic length (radius) and λ is the wavelength of the light. Rayleigh scattering applies to the case when the scattering particle is very small ($x < 1$, with a particle size $< 1/10$ of wavelength) and the whole surface re-radiates with the same phase. Because the particles are randomly positioned, the scattered light arrives at a particular point with a random collection of phases; it is incoherent and the resulting intensity is just the sum of the squares of the amplitudes from each particle and therefore proportional to the inverse fourth power of the wavelength [44]. The amount of light scattered into the solid angle with scattering angle θ_v per unit length of the LoS in the medium is called the Rayleigh angular scattering $\beta_r(\theta)$ and is given as;

$$\beta_r(\theta) = \frac{2\pi^2}{m\lambda^4} (n(\lambda) - 1)^2 (1 + \cos^2 \theta_v) \quad (2.12)$$

where $n(\lambda)$ is the wavelength dependent refractive index of the medium of scattering, m is the number density of molecules, defined as the number of molecules per unit volume. The total Rayleigh Scattering coefficient β_r is computed from angular scattering by integrating it over the solid angles as given in Equation 2.13.

$$\begin{aligned}\beta_r &= \int \beta_r(\theta) d\Omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\pi} \frac{2\pi^2}{m\lambda^4} (n(\lambda) - 1)^2 (1 + \cos^2 \theta) \sin \theta d\theta d\phi \\ &= \frac{32\pi^3 (n(\lambda) - 1)^2}{3\lambda^4 m} [m^{-1}]\end{aligned}\quad (2.13)$$

The angular scattering coefficient is retrieved from the total scattering coefficient by means of **scattering phase function** $p(\theta)$, which described how scattered light is angularly distributed.

$$p(\theta) = \frac{\beta_r(\theta)}{\int \beta_r(\theta) d\Omega} = \frac{4\pi\beta_r(\theta)}{\beta_r}\quad (2.14)$$

or by re-arranging Equation (2.14) becomes;

$$\beta_r(\theta) = \frac{\beta_r}{4\pi} p(\theta)\quad (2.15)$$

Equation (2.15) describes that the angular scattering coefficient is a product of total scattering coefficient and the phase function. The phase function of Rayleigh scattering is;

$$p(\theta) = 3/4 (1 + \cos^2 \theta)\quad (2.16)$$

Therefore the angular scattering coefficient of Rayleigh scattering becomes;

$$\beta_r(\theta) = \frac{8\pi^2 (n(\lambda) - 1)^2}{3m\lambda^4} \left[\frac{3}{4} (1 + \cos^2 \theta) \right]\quad (2.17)$$

In terms of the intensity I of light scattered by any one of the small spheres of diameter d and refractive index n from a beam of un-polarized light of wavelength λ and intensity I_o is given by equation 2.18 [45].

$$I = I_o \frac{1 + \cos^2 \theta_v}{2R^2} \left(\frac{2\pi}{\lambda} \right)^4 \left(\frac{n^2 - 1}{n^2 + 2} \right)^2 \left(\frac{d}{2} \right)^6\quad (2.18)$$

where R is the distance to the particle and θ_v is the scattering angle.

2.5.5 Mie Scattering

It describes the scattering of an electromagnetic plane wave by a homogeneous sphere. The solution takes the form of an infinite series of spherical multiple partial waves. The incident plane wave as well as the scattering field is expanded into radiating spherical vector wave functions. The internal field is expanded into regular spherical vector wave functions. By enforcing the boundary condition on the spherical surface, the expansion coefficients of the scattered field can be computed. For particles much larger or much smaller than the wavelength of the scattered light there are simple and excellent approximations that suffice to describe the behaviour of the system. But for objects whose size is similar to the wavelength, e.g., water droplets in the atmosphere, latex particles in paint, droplets in emulsions including milk, and biological cells and cellular components, a more exact approach is necessary. The formalism allows the calculation of the electric and magnetic fields inside and outside a spherical object and is generally used to calculate either how much light is scattered, the total optical cross section, or where it goes, the form factor. The notable features of these results are the Mie resonances, sizes that scatter particularly strongly or weakly.

2.5.5.1 Mie-Lorentz Approximation

Unlike Rayleigh scattering, Mie scattering theory does not come with a closed form solution. The effective cross-section and intensity can only be described in the form of a infinite expansion series [46]. An estimation of phase function of aerosol scattering can be performed by Mie-Lorentz approximation based upon the Henyey Greenstein function, which is given in Equation (2.19).

$$p(\theta) = \frac{3(1-g^2)}{2(2+g^2)} \cdot \frac{1 + \cos^2(\theta)}{(1 + g^2 - 2g \cdot \cos(\theta))^{\frac{3}{2}}} \quad (2.19)$$

where g is the symmetric factor ranging from -1.0 to 1.0 representing backscattering to forward scattering respectively.

2.5.6 Atmospheric Absorption

The effect of Rayleigh and Mie scattering in the optical spectrum is shown in Figure 2.8. Absorption of electromagnetic radiation by several gases and water vapours in the

atmosphere is also individually depicted in Figure 2.8. The combined absorption effect is also shown in the last row of the same Figure.

The absorption measurement is taken at 27°C at a humidity of 75%. Mie scattering is measured at the visibility of 23Km. CO₂ measurement is at 394 ppmv.

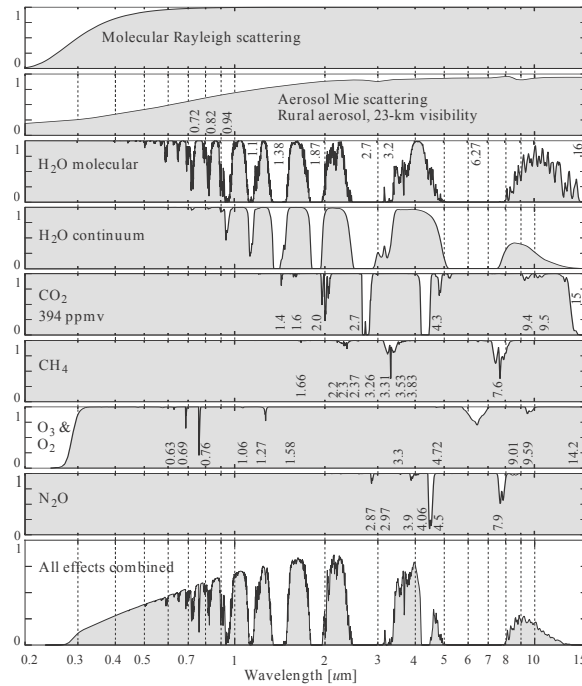


Figure 2.8: Atmospheric absorption in the electromagnetic spectrum at 27 °C.[1]

2.5.7 Fresnel Equations

When electromagnetic rays travelling in a dielectric medium with refractive index n_1 encounter the surface of another dielectric medium n_2 , at an incident angle θ_i with respect to the surface normal, part of the ray is reflected at an angle θ_r and the remaining part is transmitted at an angle θ_t . Fresnel Equations help us establish the quantitative measure of this reflection and transmission.

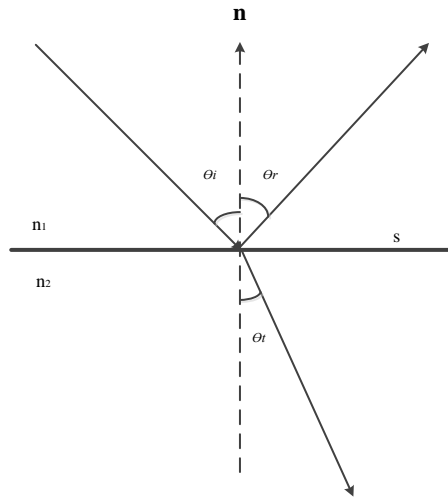


Figure 2.9: Electromagnetic Propagation at material's boundary

For a shiny reflective surface $\theta_i = \theta_r$. Transmission is governed by Snell's law of refraction is given as ;

$$n_2 \sin \theta_t = n_1 \sin \theta_i \quad (2.20)$$

2.5.7.1 Dielectric Case

The relative amount of reflected and transmitted energy is determined by the requirement that the tangential electric and magnetic fields are continuous at the interface, boundary conditions that vary depending on the polarization of the incident radiation [47]. By decomposing the incident radiation into components for which the electric fields are polarized orthogonal to the plane of incidence (s polarized or transverse electric) and in the plane of incidence (p-polarized or transverse magnetic), it becomes sufficient to understand the reflectance and transmittance properties in these two cases. Amplitude reflectivity r_s and r_p for s- and p-polarized cases, respectively, are defined as the ratio of the reflected-to-incident electric field amplitude [12]. These can be shown to be

$$r_s = \frac{n_1 \cos \theta_i - n_2 \cos \theta_t}{n_1 \cos \theta_i + n_2 \cos \theta_t} \quad (2.21)$$

or

$$r_p = \frac{n_1 \cos \theta_t - n_2 \cos \theta_i}{n_1 \cos \theta_t + n_2 \cos \theta_i} \quad (2.22)$$

Power reflectivity is defined as the ratio of the reflected-to-incident irradiance and can be determined from Equations (2.23) and (2.24) for both polarization cases as;

$$R_s(\lambda) = |r_s|^2 \quad (2.23)$$

and

$$R_p(\lambda) = |r_p|^2 \quad (2.24)$$

By conservation of energy, the corresponding power transmissivity across the interface for the two polarizations is;

$$T_s(\lambda) = 1 - R_s(\lambda) \quad (2.25)$$

and

$$T_p(\lambda) = 1 - R_p(\lambda) \quad (2.26)$$

For unpolarized incident radiation not at normal incidence, total surface reflectivity is;

$$R(\lambda) = \frac{R_s + R_p}{2} \quad (2.27)$$

and total surface transmissivity is;

$$T(\lambda) = 1 - \frac{R_s + R_p}{2} \quad (2.28)$$

2.5.7.2 Conductor Case

We explicitly define the real and imaginary parts of the complex index of refraction as

$$\tilde{N} = n + i\kappa \quad (2.29)$$

The real index n affects the phase propagation, and the imaginary index affects the extinction properties. The materials are conductive by replacing the real indices of refraction with complex indices of refraction, and by treating the angles as complex quantities

[43].

$$\tilde{N}_2 \sin \tilde{\theta}_t = \tilde{N}_1 \sin \tilde{\theta}_i \quad (2.30)$$

$$r_s = \frac{\tilde{N}_1 \cos \tilde{\theta}_i - \tilde{N}_2 \cos \tilde{\theta}_t}{\tilde{N}_1 \cos \tilde{\theta}_i + \tilde{N}_2 \cos \tilde{\theta}_t} \quad (2.31)$$

$$r_p = \frac{\tilde{N}_1 \cos \tilde{\theta}_t - \tilde{N}_2 \cos \tilde{\theta}_i}{\tilde{N}_1 \cos \tilde{\theta}_t + \tilde{N}_2 \cos \tilde{\theta}_i} \quad (2.32)$$

Equations (2.23) and (2.24) hold both for dielectric and conductor's case. The degree of polarisation (DoP) is given as,

$$DoP = \frac{R_s - R_p}{R_s + R_p} \quad (2.33)$$

In section 3.3.4 we will show the DoP of sky-radiance due to Rayleigh scattering.

2.6 Radiometry

When materials exhibit rough surfaces or interfaces, or when they contain inhomogeneity such as distributed volume scatterers, then the directionality of the propagating plane waves is significantly altered and a scattering analysis is required. To model scattering, it is necessary first to model the directional power flow of radiation which is also called radiometry. In subsequent sections, we describe some radiometric quantities with the geometrical illustration.

2.6.1 Reflectance

The reflectance of a surface is defined as the ratio between reflected radiation or radiant flux at band λ_i and the incident radiation or radiant flux at the same band. Formally it is written as Equation 2.34.

$$R = \frac{\Phi_\lambda^r}{\Phi_\lambda^i} \quad (2.34)$$

Where Φ_λ^r is reflected flux and Φ_λ^i denotes incident flux. The reflectance spectra of material is a unique entity as it depends upon the molecular structure of that material,

therefore it is also called a spectral signature.

2.6.2 Irradiance

Irradiance represents the electromagnetic power per unit area, incident on, reflected from or flowing through a surface. It is commonly given in units W/m^2 . It is formally stated as Equation 2.35.

$$E_e = \frac{\partial \Phi_e}{\partial A} \quad (2.35)$$

Where E_e is incident irradiance, $\partial \Phi_e$ is incident flux and ∂A is the area where the flux is incident upon. In case of power per unit area emitted from a source of radiation, irradiance is referred to as exitance.

2.6.3 Spectral Irradiance

Irradiance is distributed over spectral range, the distribution of irradiance per unit spectral bandwidth is termed as spectral irradiance. It is given by equation 4.21. It is commonly given in units $W/m^2.um$.

$$E_{e,\lambda} = \frac{\partial E_e}{\partial \lambda} \quad (2.36)$$

2.6.4 Solid Angle

To compute irradiance from radiance, the solid angle must be integrated into spherical coordinates as shown in Figure 2.10 where the solid angle is given as;

$$d\Omega = \frac{dA}{r^2} = \frac{rd\theta r \sin \theta d\theta}{r^2} = \sin \theta d\theta d\phi \quad (2.37)$$

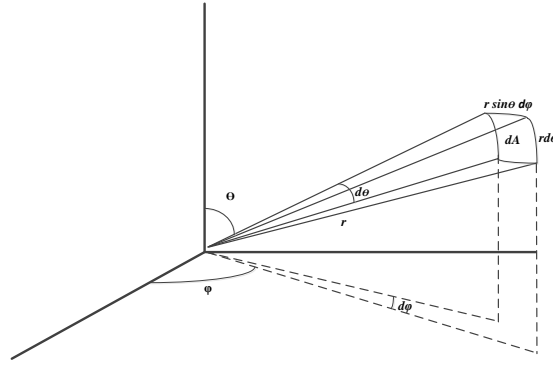


Figure 2.10: Solid Angle

where θ is the zenith and ϕ is azimuth angle of the spherical coordinate, r is the distance between two areas.

The solid angle of the sphere Ω_{sphere} is therefore given as;

$$\Omega_{sphere} = \int_0^{2\pi} \int_0^{\pi} \sin \theta d\theta d\phi = 4\pi \quad (2.38)$$

2.6.5 Point Sources

The radiant intensity I is the spatially averaged radiance over a surface. This quantity is often used to quantify the radiometric output of small or very distant point sources, such as stars. A point source radiates uniformly in all directions; therefore, the intensity is given by Equation (2.39) which is derived in the subsequent section for both point source and real objects.

$$I_e = \frac{\Phi}{4\pi} \quad (2.39)$$

The denominator in Equation 2.39 represents the fact that the solid angle of the sphere is 4π , Φ is the radiation flux, as derived in Equation (2.38).

2.6.6 Lambertian Sources

Lambert's cosine law says that the radiant intensity or luminous intensity observed from an ideal diffusely reflecting surface or ideal diffuse radiator is directly proportional to the cosine of the angle θ between the direction of the incident light and the surface normal. Moreover, radiance is independent of angles.

$$\Phi = LA \int_0^{2\pi} \int_0^{\pi/2} \cos \theta \sin \theta d\theta d\phi \quad (2.40)$$

$$E = L \int_0^{2\pi} \int_0^{\pi/2} \cos \theta \sin \theta d\theta d\phi \quad (2.41)$$

Although the total solid angle of the hemisphere is 2π , due to the cosine factor the relationship between radiance and irradiance radiated from a lambertian source reduces to,

$$E = \pi L \quad (2.42)$$

2.6.7 Bi-directional Reflection Distribution Function

Bi-directional Reflection Distribution Function is a characteristic scattering property of a material. Unlike irradiance, BRDF depends on the spherical coordinate components of the incident irradiance and reflected radiance. Mathematically it is given in Equation 2.43. The geometric entities are illustrated in Figure 2.43.

$$f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) = \frac{dL_r(\theta_r, \phi_r)}{dE(\theta_i, \phi_i)} \text{sr}^{-1} \quad (2.43)$$

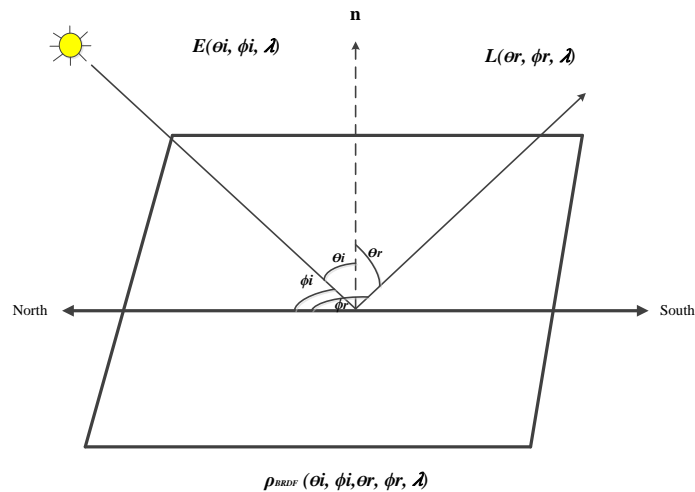


Figure 2.11: Bi-directional Reflection Distribution Function

where λ denotes wavelength, E is the incident irradiance and L_r is the reflected radiance. θ_i is the incident irradiance's zenith or according to illustration solar zenith, θ_r is the reflected radiance zenith with respect to the surface normal. The incident irradiance azimuth with respect to the North is denoted as ϕ_i , while ϕ_r is azimuth angle of the reflected radiance. Unit of BRDF is sr^{-1} .

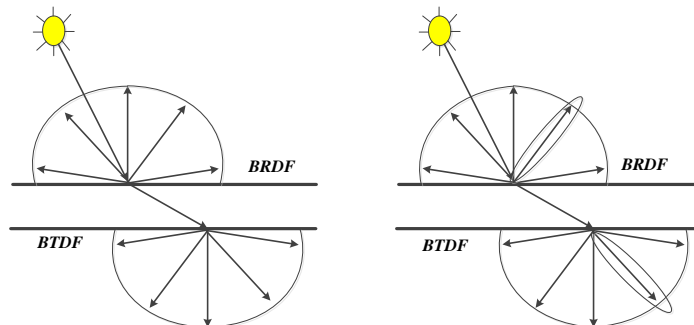


Figure 2.12: Bi-directional Reflection/Transmission Distribution Function

Similar to BRDF, Bi-directional Transmission Distribution Function is a characteristic transmission property of a material. Both BRDF and BTDF are shown in Figure 2.12 for a Lambertian surface (left), and a generic surface that possess both Lambertian and specular properties.

There are several BRDF models for the Visual and Near-InfraRed (VNIR) spectrum such as Phong (fit of cosine), Ward (fit of a Gaussian shape), Ashikhmin, Cook–Torrance (specular microfacet) and Torrance–Sparrow etc. In the subsequent section we will discuss the simplistic Phong model which is included in CHIMES simulator. We also show a comparison of Phong BRDF model for given parameters and a corresponding real-world BRDF of materials.

2.6.7.1 Phong BRDF

Phong models BRDF by mean of three parameters i.e. ρ_d which is the Lambertian reflection constant, n determines the angular deviation of the lobe, and ρ_s determines the peak value or ‘strength’ of the lobe. A more specific version of reflection geometry to illustrate the factors of Phong BRDF is given in Figure 2.13. Here θ_i and θ_r are incident and reflected ray angle to the surface normal in a perfectly specular surface. The angle

α is illustrated as BRDF specific reflection angle's deviation from θ_r , as shown in Figure 2.13. The model is given in Equation (2.44).

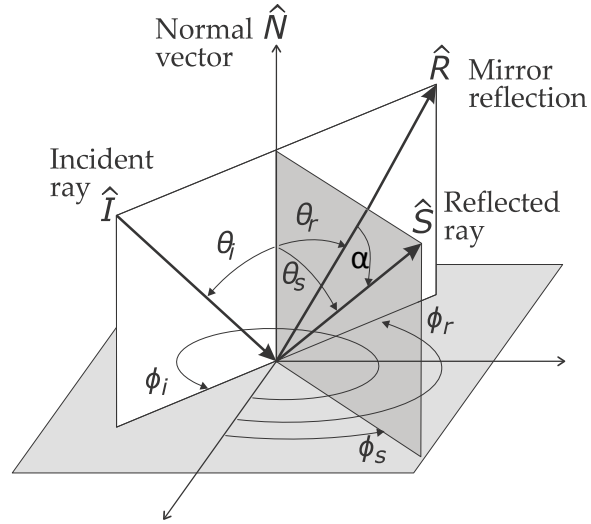


Figure 2.13: Reflection Geometry illustrating parameters of Phong BRDF.

$$f_{r, \text{Phong}} = \frac{\rho_d}{\pi} + \frac{\rho_s(n+1) \cos^n \alpha}{2\pi \cos \theta_i} \quad (2.44)$$

Varying the values of these parameter we may get different fits of cosines as shown in Figure 2.14. Similar real-world BRDF of various materials are also shown in the bottom row of the same Figure.

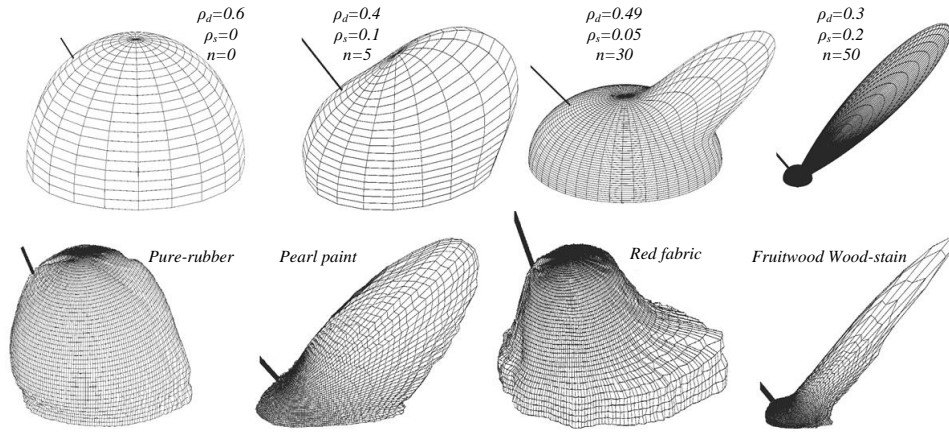


Figure 2.14: First row: BRDF model created by Phong equation having variable values for key paramters [1]. Second row: Measured BRDF of know material recorded in MERL benchmark dataset [2, 3].

2.6.8 Summary

Radiometric quantities and their relationship with each other are summarized in Table 2.2.

Table 2.2: Radiometric quantities, symbols, their relationship and units [12].

Quantity	Symbols	Relationship	Units
Irradiance	E	$E = \frac{\partial \Phi}{\partial A} = \int E_{\lambda} d\lambda$	W/m^2
Spectral irradiance	E_{λ}	$E_{\lambda} = \frac{\partial \Phi}{\partial A \partial \lambda} = \int L_{\lambda} d\Omega$	$\text{W}/\text{m}^2\text{m}$
Radiance	L	$L = \frac{\partial \Phi}{\partial A \partial \Omega} = \int L_{\lambda} d\lambda$	$\text{W}/\text{m}^2\text{sr}$
Spectral radiance	L_{λ}	$L_{\lambda} = \frac{\partial \Phi}{\partial A \partial \Omega \partial \lambda} = \int L dA$	W/sr
Power	Φ	$\Phi = \int E dA = \int I d\Omega$	W

2.7 At-sensor Radiance Models and Components

The observed radiance by imaging systems are accumulated by several individual radiance components as labelled in Figure 2.15 from A to I. These components include reflected sunshine (A), thermal emission from target (D), reflected sky shine (B, E), reflected background radiance (G, H), path radiance (C, F) and upwelled scattered adjacent radiance (I). As our focus in this research is on upwelled scattered adjacent radiance therefore

discussion regarding several models are described separately in section 2.9.

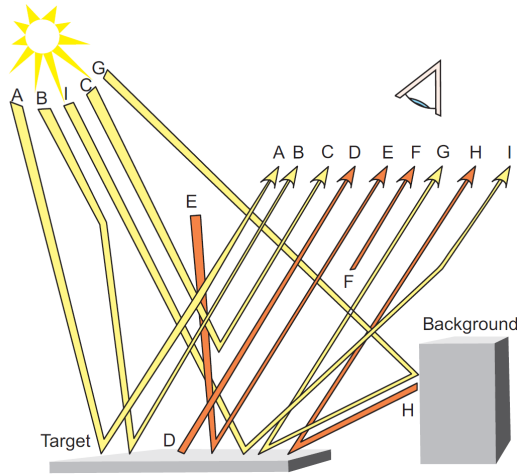


Figure 2.15: DIRSIG big equation components [4, 5]. Yellow rays are emitted by the sun while orange rays are emitted from various objects in the environment. Ray "A" represent direct reflected radiance, "B" shows diffused reflected radiance, "C" is scattered into the sensor's LoS. Ray "D" is emitted by the target object. "E" is emitted from the environment and reflected from the target object. "F" is in-scattered from the atmosphere into sensor's LoS. "G" is background reflected. "H" is emitted by background and reflected off the target object and I is reflected and scattered into the sensor LoS.

Figure 2.15 shows light rays that are emitted by the primary light source (sun) and reflected off the target or atmosphere by yellow colour. The thermal radiations due to the atmosphere, target or background object is shown in orange colour. Rays that are not reflected off the target such as "B", "F" and "I" constitute the upwelled scattered additive radiance. Estimating these components require understanding of aerosol scattering and the underlying phase function model for the specific scene, which is complicated phenomena and it is typically estimated by DISORT algorithm. Contrarily, direct and diffuse reflected radiance is relatively easier to estimate and require geometric understanding about emitter's positioning. In this research our we focus on the better estimation of upwelled scattered radiance.

2.7.1 Solar irradiance

The incident (**downwelled**) solar irradiance E_i is given in Equation (2.45), where E_{TOA} is exoatmospheric solar irradiance which is sometimes also called as Top of Atmosphere

irradiance, θ_i is the incident solar zenith angle with respect to the surface normal (n), as shown in Figure 2.11.

$$E_i = E_{TOA} \cos \theta_i \tau_1 \quad (2.45)$$

τ_1 is the transmission along the incident Line Of Sight (LoS) from source to the target object. Individual components of E_i are shown in Figure 2.16.

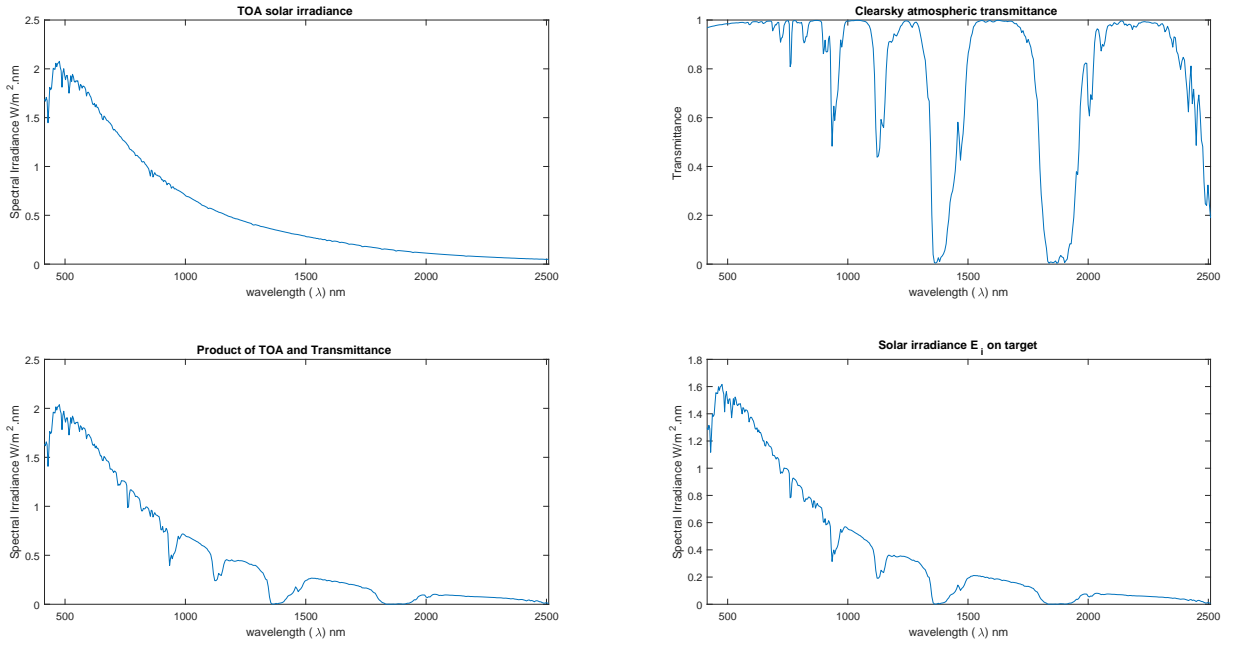


Figure 2.16: ToA solar irradiance, clear-sky atmosphere, solar irradiance at target

The irradiance and transmission quantities in Figure 2.16 are extracted from MODTRAN for a "No-aerosol" setting which represents a clear-sky atmospheric absorption. Solar irradiance E_r , incident upon a target on the ground is also shown.

The reflected solar irradiance E_r (upwelled) is given as;

$$E_r = f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) E_{TOA} \cos \theta_i \tau_1 \tau_2 \rho \quad (2.46)$$

where τ_2 is the transmission from the target object to the sensor, as shown in Figure 2.11 and $f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)$ is the BRDF of the target object.

2.7.2 Sky Irradiance

The sky-shine is a consequence of sunlight scattering by the atmosphere. If the volume of medium confined inside the subtended solid angle cone is assumed as dV , as shown in Figure 2.17 then the relationship is given as;

$$dV = dA dr = d\Omega r^2 dr \quad (2.47)$$

where dA is the area subtended by the solid angle $d\Omega$ at distance r . The incident irradiance into this volume becomes;

$$E_v = E_{TOA} \tau_{L1} \quad (2.48)$$

E_v represents the incident irradiance on the volume V . It should be noted that unlike incident direct solar irradiance, the incident sky irradiance on the **target** has two transmission components in the downwelling ring i.e. τ_{L1} and τ_{L2} . τ_{L1} along the LoS from the sun to the scattering event point and τ_{L2} from the scattering event point to the target.

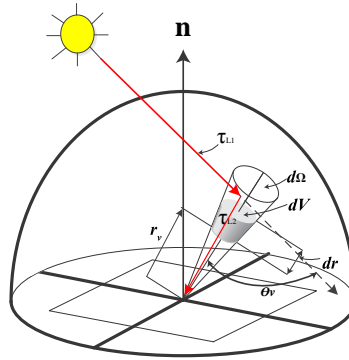


Figure 2.17: solar downwelled scattered radiance.

The radiant intensity scattered into the volume dV is therefore given as;

$$dI_v = E_v \beta_{sca}(\theta_v) dV \quad (2.49)$$

Placing value of E_v from Equation (2.48) into Equation (2.49), we get;

$$dI_v = E_{TOA} \tau_{L1} \beta_{sca}(\theta_v) dV \quad (2.50)$$

Value of dV from (2.48) is placed into Equation (2.50) to get;

$$dI_v = E_{TOA} \tau_{L1} \beta_{sca}(\theta_v) d\Omega r_v^2 dr \quad (2.51)$$

If the incident volumetric intensity is perpendicular to the target then Equation (2.51) becomes;

$$dE_d = \frac{E_{TOA} \tau_{L1} \tau_{L2} \beta_{sca}(\theta_v) d\Omega r_v^2 dr}{r_v^2} \quad (2.52)$$

The spectral radiance along the downwelled scattered ray at target can be expressed as;

$$dL_d(\theta, \phi) = \frac{E_{TOA} \tau_{L1} \tau_{L2} \beta_{sca}(\theta_v) d\Omega dr}{d\Omega} \quad (2.53)$$

where the downwelling sky radiance is a function of spherical coordinates zenith and azimuth angles. The total downwelling radiance is computed by integrating dL_{sky} over the distance r from target to the top of the atmosphere.

$$L_{sky}(\sigma, \phi) = E_{TOA} \int \tau_{L1} \tau_{L2} \beta_{sca}(\theta_v) dr \quad (2.54)$$

The total downwelled spectral radiance is computed from angular radiance by integration over the hemisphere above the target, which is given as;

$$E_{sky} = \int L_{sky}(\sigma, \phi) \cos \theta d\Omega = \int_{\phi=0}^{2\pi} \int_{\theta=0}^{\frac{\pi}{2}} L_{sky}(\theta, \phi) \cos \theta \sin \theta d\theta d\phi \quad (2.55)$$

2.7.3 Background Reflected Radiance

Radiance incident on a target that is reflected by the background is an important incident light constituent. Suppose that a target is placed on an extremely large ground that has no obstruction, then the sky-view would be unoccluded by background and would ideally be the entire hemisphere. The sky-view, therefore, help us to quantify the background objects and their effect on incident flux.

The **Sky-view factor** quantifies the fraction of visible hemisphere above the target, it is typically denoted as F . The background is quantified as a complementary factor of the sky-view i.e. $1 - F$, as the maximum value of F is 1.0. In our simulations, the

heightmap is given as an image where height represents intensity. The simulator then performs interpolation and creates a 3D geometry based on the input map. Therefore we will follow the roof sky-view model in this section.

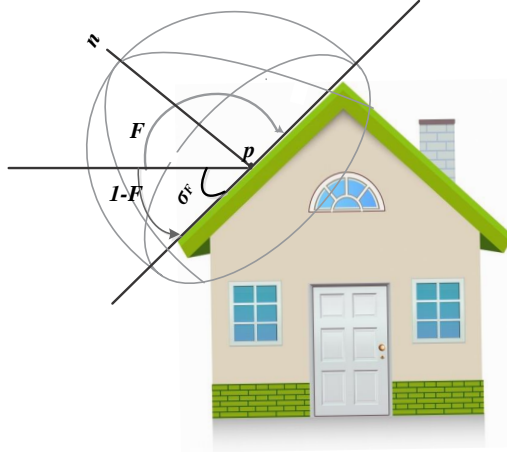


Figure 2.18: Sky-view factor for a roof slope geometry

If solid angle $d\Omega_b$ is blocked by background objects then the background factor is normalized with the solid angle of the hemisphere, so F_{back} is given as;

$$F_{back} = \frac{\int d\Omega_b}{2\pi} \quad (2.56)$$

and F_{sky} is;

$$F_{sky} = 1 - F_{back} \quad (2.57)$$

The sky-view factor F_{sky} for the point p on the roof 2.18 is given as;

$$F \cong 1 - \frac{\int_{\phi=0}^{\pi} \int_{\sigma-\frac{\pi}{2}}^{\frac{\pi}{2}} \sin \sigma d\sigma d\phi}{2\pi} = 1 - \frac{1}{2} \cos \left(\frac{\pi}{2} - \sigma_F \right) \quad (2.58)$$

After calculating the sky-view factor, we may now proceed to find the irradiance incident upon target from the reflection of background. The background reflected irradiance $E_{background}$ is expressed as;

$$E_{background} = \iint L_{background}(\sigma, \phi) \cos \sigma \sin \sigma d\sigma d\phi \approx (1 - F)L_{background}\pi \quad (2.59)$$

The background irradiance is approximated for a Lambertian surface in the Equation (2.59).

2.7.4 Path Radiance

In section 2.7.2 we found the incident irradiance and radiance on a target from the sky, i.e. sunlight scattered by the atmosphere, into targets line of sight. Path radiance is a similar phenomenon where light is scattered into the sensor's/camera's LoS instead of the target's LoS. It is illustrated in Figure 2.19. The figure shows that an observer (sensor/camera) is observing the target and the scattering event happens in the subtended solid angle volume but its direction is towards the observer. As we are interested in the observers LoS, therefore, we will compute the path radiance scattered upwelled.

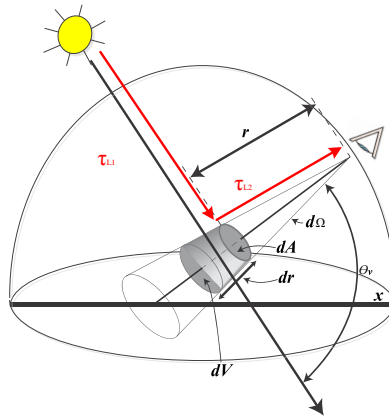


Figure 2.19: Solar radiance scattered towards the LoS of the sensor without coming in contact with the target. (path radiance)

As shown in Figure 2.19, path radiance does not come in contact with ground and therefore does not bring any information about the target. Analogous to the downwelled scattered radiance the path radiance is expressed as;

$$L_{path}(\sigma, \phi) = E_{TOA} \int \tau_{L1} \tau_{L2}' \beta_{sca}(\theta_u) dr \quad (2.60)$$

In comparison with Equation (2.54), the terms τ_{L2}' and r have upwelled direction. Accumulating all components discussed above the at-sensor radiance can be expressed as Equation (2.61). Although thermally emitted radiance and thermal scattering are not considered in the construction of this simulator thermal upwelled scattering is also included in the equation to display a comprehensive form of Radiative Transfer equation in Equation (2.61).

$$L_{sensor} = \underbrace{\int_{\Omega} f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) E_{TOA} \cos \theta_i \tau_1 \tau_2 \rho d\omega}_{\text{solar reflected}} + \underbrace{\varepsilon_o L(T_o) \tau_1}_{\text{thermally emitted}} + \underbrace{E_{TOA} \int \tau_{L1} \tau_{L2} \beta_{op}(\theta_v) dr \rho}_{\text{sky radiance}} +$$

$$\underbrace{\iint L_{background}(\sigma, \phi) \cos \sigma \sin \sigma d\sigma d\phi \rho}_{\text{background reflected radiance}} + \underbrace{E_{TOA} \int \tau_{L1} \tau_{L2}' \beta_{op}(\theta_{uo}) dr}_{\text{path radiance}} +$$

$$\underbrace{\varepsilon E(T) \int \tau_{L2}' \beta_{th}(\theta_{ut}) dr}_{\text{thermally scattered}} \quad (2.61)$$

Where β_{op} denotes optical scattering coefficient where β_{th} stands for thermal ones. The phenomena of scattering where light scattered into the volume dV are called in-scattering, similarly when light is scattered out of the dV then it is out-scattering. The upwelled scattered adjacent radiance, however, is included in the section 2.9.

2.8 Monte-Carlo Integration

In this section, We will provide background theory for Monte-Carlo Integration to describe the at-sensor equation using Monte-Carlo Integral. Before going in details about Monte-Carlo methods we will briefly define relevant probability theory to establish the necessary background.

The at-sensor radiance equation is a higher dimensional equation having discontinuities, therefore it is difficult to find an analytical solution of the equation. The trapezoidal integration typically works well with the continuous smooth and lower dimensional functions, therefore it is not suitable to the rendering process.

2.8.1 Cumulative Distribution Function

The Cumulative Distribution Function of a random variable X is smaller than or equal to a value x ;

$$P(x) = \Pr\{X \leq x\} \quad (2.62)$$

The probability density function (pdf) is defined as;

$$\text{pdf}(x) = \frac{dP(x)}{dx} \quad (2.63)$$

2.8.1.1 Expected Value and Variance

The expected value of a random variable $X \in D$ is the average value over the distribution of values $\text{pdf}(x)$:

$$E\{X\} = \int_D x \text{pdf}(x) dx \quad (2.64)$$

Therefore, the expected value of a function $f(x)$ is;

$$E\{f(X)\} = \int_D f(x) \text{pdf}(x) dx \quad (2.65)$$

Variance of the function $f(x)$ is therefore;

$$V\{f(X)\} = E\{(f(X) - E\{f(X)\})^2\} \quad (2.66)$$

which implies;

$$V\{f(X)\} = E\{(f(X))^2\} - (E\{f(X)\})^2 \quad (2.67)$$

The law of large numbers states when the number of samples become very large then the cumulative density function approaches the expected value of the function;

$$\Pr\left\{\frac{1}{N} \sum_{j=1}^N f(X_j) \rightarrow E\{f(X)\}\right\} = 1 \text{ for } N \rightarrow \infty \quad (2.68)$$

Monte-Carlo's approximate of an arbitrary integral by N samples is expressed as;

$$F = \int_D f(x)dx = \int_D \frac{f(x)}{\text{pdf}(x)} \text{pdf}(x)dx = E \left\{ \frac{f(X)}{\text{pdf}(X)} \right\} \quad (2.69)$$

Therefore by applying the law of large numbers the Monte Carlo approximation of integral becomes;

$$F_N = \frac{1}{N} \sum_{j=1}^N \frac{f(X_j)}{\text{pdf}(X_j)} \quad (2.70)$$

2.8.2 Monte-Carlo Convergence

Standard deviation of F_N is given as;

$$\sigma_{F_N} = (V \{F_N\})^{1/2} = \left(V \left\{ \frac{1}{N} \sum_{j=1}^N Y_j \right\} \right)^{1/2} \quad (2.71)$$

$$\sigma_{F_N} = \left(\frac{1}{N^2} \sum_{j=1}^N V \{Y_j\} \right)^{1/2} \quad (2.72)$$

$$\sigma_{F_N} = \left(\frac{1}{N} V \{Y\} \right)^{1/2} = \frac{1}{\sqrt{N}} \sigma_Y \quad (2.73)$$

Where σ_{F_N} is the standard deviation of of the estimator and σ_Y is the standard deviation of Y. Monte Carlo integration convergence rate has a time complexity with the bound of $(O(1/\sqrt{N}))$ with an arbitrary dimension and is independent of smoothness of integrant.

2.8.3 Monte-Carlo Sampling Methods

To compute Monte Carlo integration it is necessary to be able to draw samples from a given probability distribution. In this section, we will describe some popular sampling schemes for MC integration.

2.8.3.1 Inversion

The inversion method employs one of the more uniform random variables and maps them to the random variable of the desired probability distribution. Suppose, for a given probability distribution the Cumulative Distribution Function (CDF) is $P(x)$ which peak at the highest probability (1.0) i.e. monotonically non-decreasing. Now a uniform random

variable η is projected on the CDF function to get the probability from the CDF. For a continuous CDF, function projection may be computed by inverting the CDF and evaluating the inverse at η . Therefore, this method is called the inversion method. A pseudo-code of inversion algorithm is shown in Table 1 [48].

Algorithm 1 Drawing samples from arbitrary PDF by inversion method

1: Compute CDF :

$$P(x) = \int_{-\infty}^x \text{pdf}(x') dx' \quad (2.74)$$

2: Find inverse CDF :

$$P^{-1}(x) \quad (2.75)$$

3: Get a uniformly distributed random number ξ :

$$\xi \in [0, 1] \quad (2.76)$$

4: Compute Sample X_i :

$$X_i = P^{-1}(\xi) \quad (2.77)$$

2.8.3.2 Rejection Sampling

In some cases where it is not possible to integrate $f(x)$ to compute its PDF or it is not possible to analytically invert their CDF then Rejection Method is employed. Suppose we draw samples from function $f(x)$ and we have a PDF $p(x)$ that satisfies $f(x) < cp(x)$ for some constant c and suppose that we can sample from p , then we will select the sampled X if $\eta < f(X)/cp(X)$, otherwise it will be rejected.

2.8.3.3 Importance sampling

Importance sampling is a variance reduction technique which states that the Monte Carlo estimator F_N converges quickly if samples are taken from a PDF **similar** to the function $f(x)$. Figure 2.20 shows three different PDFs for computing F_N . In the middle, we place a uniform distribution and show that PDF similar to $f(x)$ will produce variance reduction

over the uniform one. Contrarily, the first pdf will have higher variance compared to other two PDFs of Figure 2.20.

$$F_N = \frac{1}{N} \sum_{j=1}^N \frac{f(X_j)}{\text{pdf}(X_j)} \quad (2.78)$$

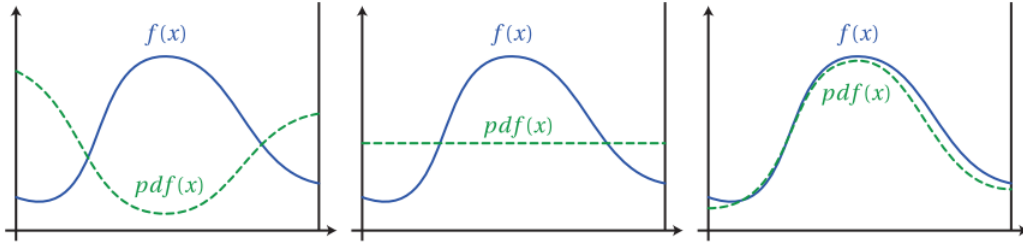


Figure 2.20: Estimating $f(x)$ by $\text{pdf}(x)$: (a) bad pdf (b) uniform pdf (c) similar pdf

The idea is to focus on the part where integrand is relatively high to compute good estimate efficiently. For example, in terms of our downwelling at-sensor radiance equation, the solar irradiance depends heavily on the cosine of solar zenith, therefore, the value of direct solar irradiance is lower along the horizon where the incident angle approaches to be perpendicular to the surface normal. Therefore sampling should be given **higher importance** to the incident angles where $\cos \theta$ is higher.

Monte-Carlo method estimates the integral of the function of form $\int f(x)dx$. Our at-sensor radiance Equation has integral of a product of more functions, such as $\int f(x)g(x)dx$. In order to address such integrals the implementer may establish a strategy for both $f(x)$ and $g(x)$.

consider that the surface on which L_r is incident has a perfectly specular BRDF. Due to this assumption, the value of integrand will be zero for all directions except the specular (smooth reflection's) one. If the light source is large area light such as environment map large number of directions needs to be sampled, therefore PDF value decreases. In such a case even if the sampled direction is specular and integrand value is higher it is divided by a smaller PDF value.

Multiple Importance Sampling (MIS) draws samples from multiple sampling distribution with the hope that at least one of them will match the shape of the integrand. MIS provides a strategy for weighing the samples such that it can eliminate large variance transitions due to difference between integrand value and sampling densities. Suppose if

two sampling distributions pdf_f and pdf_g are used to estimate $\int f(x)g(x)dx$ the estimation by means of MIS is given as;

$$\frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(X_i)g(X_i)w_f(X_i)}{p_f(X_i)} + \frac{1}{n_g} \sum_{j=1}^{n_g} \frac{f(Y_j)g(Y_j)w_g(Y_j)}{p_g(Y_j)} \quad (2.79)$$

where n_f and n_g are number of samples from PDF of function f and g respectively. Similarly, w_f and w_g are weighing functions for f and g , respectively. A potential alternate of computing weight function is **balanced heuristic** which is given as;

$$w_s(x) = \frac{n_s p_s(x)}{\sum_i n_i p_i(x)} \quad (2.80)$$

Suppose sample X is drawn from pdf_f at the point where $\text{pdf}_f(x)$ is low and it is closer to $f(x)$ then $f(x)$ is also lower. However, if $g(x)$ is a relatively higher value then the importance sampling in Equation (2.81) will have a very large value, as $\text{pdf}_f(x)$ is small and will yield a higher variance.

$$\frac{f(X)g(X)}{p_f(X)} \quad (2.81)$$

In terms of MIS together with Balanced heuristic, the contribution of X will be;

$$\frac{f(X)g(X)w_f(X)}{p_f(X)} = \frac{f(X)g(X)n_f p_f(X)}{p_f(X)(n_f p_f(X) + n_g p_g(X))} = \frac{f(X)g(X)n_f}{n_f p_f(X) + n_g p_g(X)} \quad (2.82)$$

In this case if pdf_g is a reasonable match for $g(x)$ then the denominator won't be too small due to the term $n_g p_g(X)$ and would eliminate transients in the variance.

[49] propose that powers in balanced heuristic, also called as power heuristic is in practice reduces variance more than the balanced one.

$$w_s(x) = \frac{(n_s p_s(x))^\beta}{\sum_i (n_i p_i(x))^\beta} \quad (2.83)$$

[49], also suggests the value of $\beta=2$ produce optimal results based on empirical conclusions.

2.8.4 Monte Carlo Estimation of At-Sensor Radiance

In this section, we will express the components of at-sensor radiance Equation (2.61) in the form of Monte Carlo estimation with multiple importance sampling. We will also express the distance to scattering event and upwelled scattered radiance terms in Monte Carlo formulation.

2.8.4.1 Direct Solar

Suppose we have the direct solar term to solve by importance sampling, Equations (2.84),(2.85) and (2.86) express solar reflected term, it's Monte Carlo estimation and multiple importance sampling form of the equation, respectively.

$$L_r = \overbrace{\int_{\Omega} f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) E_{TOA} \cos \theta_i \tau_1 \tau_2 \rho d\omega}^{\text{solar reflected}} \quad (2.84)$$

$$= \frac{1}{N} \sum_{j=1}^N \frac{f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) L_i(x, \vec{\omega}_i) \cos \theta_i}{\text{pdf}(\vec{\omega}_i)} \quad (2.85)$$

$$= \frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) L_i(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) w_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))}{p_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))} + \frac{1}{n_{Li}} \sum_{j=1}^{n_{Li}} \frac{f(\vec{\omega}_i) L_i(\vec{\omega}_i) w_{Li}(\vec{\omega}_i)}{p_{Li}(\vec{\omega}_i)} \quad (2.86)$$

[49] reduced L_r approximation to Equation (2.87), as the second term represent one of the direction vector of the emitter.

$$= \frac{1}{n_f} \sum_{i=1}^{n_f} \frac{f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) L_i(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) w_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))}{p_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))} \quad (2.87)$$

To solve the MIS form we need to understand the direction sampling. **Direction sampling** in the cosine weighted hemisphere for pdf(ω) is given as;

$$\text{pdf}(\vec{\omega}) = \cos \theta \pi \quad (2.88)$$

$$\text{pdf}(\theta, \phi) = \cos \theta \sin \theta / \pi \quad (2.89)$$

The marginal and conditional probability function for zenith and azimuth angles there-

fore expressed in Equations (2.90) and (2.91), respectively.

$$\text{pdf}(\theta) = \int_0^{2\pi} \frac{\cos \theta}{\pi} \sin \theta d\phi = 2 \cos \theta \sin \theta \quad (2.90)$$

$$\text{pdf}(\phi|\theta) = \frac{\cos \theta \sin \theta / \pi}{2 \cos \theta \sin \theta} = \frac{1}{2\pi} \quad (2.91)$$

The CDF of marginal density function is therefore;

$$P(\theta) = 2 \int_0^\theta \cos \theta' \sin \theta' d\theta' = 2 \int_1^{\cos \theta} (-\cos \theta') d\cos \theta' = 1 - \cos^2 \theta \quad (2.92)$$

$$P(\phi|\theta) = \phi / (2\pi) \quad (2.93)$$

The direction sampling strategy can be defined as a function of uniform random variable by Equation (2.94).

$$\vec{\omega} = (\theta, \phi) = \left(\cos^{-1} \sqrt{\xi_1}, 2\pi\xi_2 \right) \quad (2.94)$$

Radiance transmitting through a homogeneous medium (having a uniform extinction co-efficient) is influenced by its transmittivity and is expressed as;

$$L_{tr}(R) = e^{-\mu_e R} L_o \quad (2.95)$$

where L_o is the radiance at originating point R_o and $L_{tr}(R)$ is the transmitted radiance at a distance of R from the R_o .

2.8.4.2 Sky Radiance

Sky radiance has two transmission legs as given in Equation (2.96), transmission of both legs are combined for a homogeneous medium with extinction coefficient (μ_e) and is given in Equation (2.97).

$$L_{sky} = E_{TOA} \overbrace{\int \tau_{L1} \tau_{L2} \beta_{op}(\theta_v) dr}^{\text{sky radiance}} \quad (2.96)$$

$$L_{\text{sky}} = \frac{1}{N} \sum_{j=1}^N \frac{T_r(R, R_o) \beta_{op}(\theta_v)}{\text{pdf}(R)} \quad (2.97)$$

Sampling of $\text{pdf}(R)$, which is an exponential function is given as;

$$\text{pdf}(R) = \beta_{op} T_r(R, R_o) = \beta_{op} e^{-\mu_e(R-R_o)} \quad , \quad R - R_o = -\frac{\ln(\xi_j)}{\mu_e} \quad (2.98)$$

Where $R - R_o$ is the distance through the transmission legs and during ray tracing it is the distance to the next scattering event. R_o is the originating point of the distance and R is the terminal point. The estimate of L_{sky} by MIS is expressed in Equation (2.99).

$$L_{\text{sky}} = \frac{1}{N_f} \sum_{j=1}^{N_f} \frac{T_r(R, R_o) \beta_{op}(R, R_o) w_f(R, R_o)}{\text{pdf}(R, R_o)} + \frac{1}{N_g} \sum_{j=1}^{N_g} \frac{T_r(\theta_v) \beta_{op}(\theta_v) w_g(\theta_v)}{\text{pdf}(\theta_v)} \quad (2.99)$$

$\text{pdf}(\theta_v)$ is computed by Henyey Greenstein function which is regulated by a symmetry operator g whose value is between -1.0–1.0. Scattering properties regulated by parameters are;

- $g = -1$ total backscattering
- $g = 0$ isotropic scattering
- $g = 1$ total forward scattering.

$$p(\theta_v) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 - 2g \cos \theta_v)^{3/2}} \quad (2.100)$$

A simple way of multiple importance sampling the phase function is given in Equations (2.101).

$$\cos \theta_k = \begin{cases} \frac{1}{2g} \left[1 + g^2 - \left(\frac{1-g^2}{1-g+2g\xi_k} \right)^2 \right] & \text{for } g \neq 0 \\ 2\xi_k - 1 & \text{for } g = 0 \end{cases} \quad (2.101)$$

We have formulated all the components to measure the sky radiance by multiple importance sampling in Equations (2.97–2.101).

2.8.5 Existing Simulators At-sensor equations

In this section we will discuss details about radiative transfer (RT) equations of light-transport based models mentioned in previous section. This include (i) DIRSIG, (ii) OS-SIM, and (iii) CameoSim, that commonly use adjacency-effect model based on DISORT. Due to the deviation of adjacency effect model of DISORT in ATCOR, we will include ATCOR's RT equation for assessment in the subsequent section. The rendering mechanism of these simulators also varies, for example, DIRSIG version 4 employs Whitted-style ray tracing, CameoSim renders by radiosity and CHIMES follow path-tracing strategy for computing the at-sensor radiance. The diverse nature of methods applied on HSI simulator require blend of graphics, remote sensing, sampling, reconstruction filtering knowledge, so a background of computer graphics may not be expected from reader with background in remote sensing community [18]. We therefore subsequently illustrate the differences between ray-tracing and path-tracing, in brief.

2.8.5.1 DIRSIG

The DIRSIG RT equation is referred to as the Big Equation [20],[28]. Radiance components of Big Equation are shown as different types of photons labelled from A to I, as illustrated in Figure 2.21. Assuming the BRDF of target object to be Lambertian, the big equation takes the form of equation 2.102. According to [5] "I" type photons can be grouped with C type photons if the average albedo of the scene is slowly varying, which is common [1]. DIRSIG follows an image-chain differential thermodynamic model to calculate the temperature of background objects in the scene. It accounts for both optical and thermal shadow caused by full or partial occlusion due to opaque or transmissive objects. During rendering interaction of rays with medium and transmittance loss is also accounted [18].

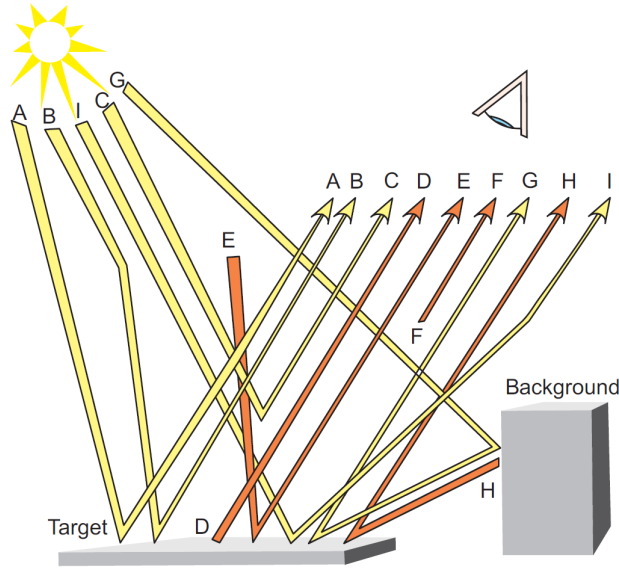


Figure 2.21: DIRSIG big equation components [4, 5]

$$\begin{aligned}
 L(\lambda) = \tau_2(\lambda) \left\{ \underbrace{\frac{E'_s(\lambda)}{\pi} \tau_1(\lambda) \rho(\lambda) \cos \sigma}_A \right. &+ \underbrace{\varepsilon(\lambda) L_T(\lambda)}_D + \rho_d(\lambda) F \left[\underbrace{\frac{E_{ds}(\lambda)}{\pi}}_B + \underbrace{\frac{E_{d\varepsilon}(\lambda)}{\pi}}_E \right] + \\
 &\left. \underbrace{\rho_d(\lambda) (1 - F) \left[\underbrace{L_{bs}(\lambda)}_G + \underbrace{L_{b\varepsilon}(\lambda)}_H \right]}_{\text{reflected background}} \right\} + \underbrace{L_{us}(\lambda)}_{C+I} + \underbrace{L_{u\varepsilon}(\lambda)}_F . \quad (2.102)
 \end{aligned}$$

reflected sunshine
thermally emitted
reflected sky

Table 2.3: Symbols in Equation 2.102 and their meaning, λ shows that quantities are function of wavelength.

Symbol	Meaning	Symbol	Meaning
$E'_s(\lambda)$	Exoatmospheric irradiance	1-F	Fraction of hemisphere above target that is background
$\cos \sigma$	Solar zenith angle w.r.t target normal	$E_{ds}(\lambda)$	Downwelled solar irradiance
$\tau_1(\lambda)$	Transmittance along sun-target path	$E_{d\varepsilon}(\lambda)$	Downwelled self-emitted radiance from the atmosphere
$\tau_2(\lambda)$	Transmittance along target-sensor path	$r_d(\lambda)$	Target diffuse reflectance
$r(\lambda)$	Target reflectance	$L_{bs}(\lambda)$	Background reflected solar radiance onto target
$\varepsilon(\lambda)$	Target emissivity	$L_{b\varepsilon}(\lambda)$	Background self emitted irradiance onto target
$L_T(\lambda)$	Self emitted radiance from target at temperature T	$L_{us}(\lambda)$	Upwelled solar irradiance
F	Fraction of hemisphere above target that is sky	$L_{u\varepsilon}(\lambda)$	Upwelled self-emitted radiance

DIRSIG uses MODTRAN for its atmospheric processing. It also creates a sky-dome

environment map for illumination which is created by MODTRAN based lookup tables [50]. The texture applied to geometry are used as material and thermodynamic database identifier [4].

2.8.5.2 OSSIM

Scene geometry is constructed as a three dimensional complex hull consisting of flat, convex hull of polygons. Each geometric primitive is assigned spatial texture and radiometric properties. It can be opaque, transparent or heterogeneous medium such as countermeasure flare or aircraft plume. OSSIM has a comprehensive RT equation which is given in Equation (2.103). Symbols and their meanings are listed in Table 2.4. All the terms in Equation (2.103) are function of λ , denoting the spectral nature of processing for sources such as, illumination, plumes and atmosphere. Atmospheric processing employs all capabilities of MODTRAN, which is an integral part of the software. Scene geometric

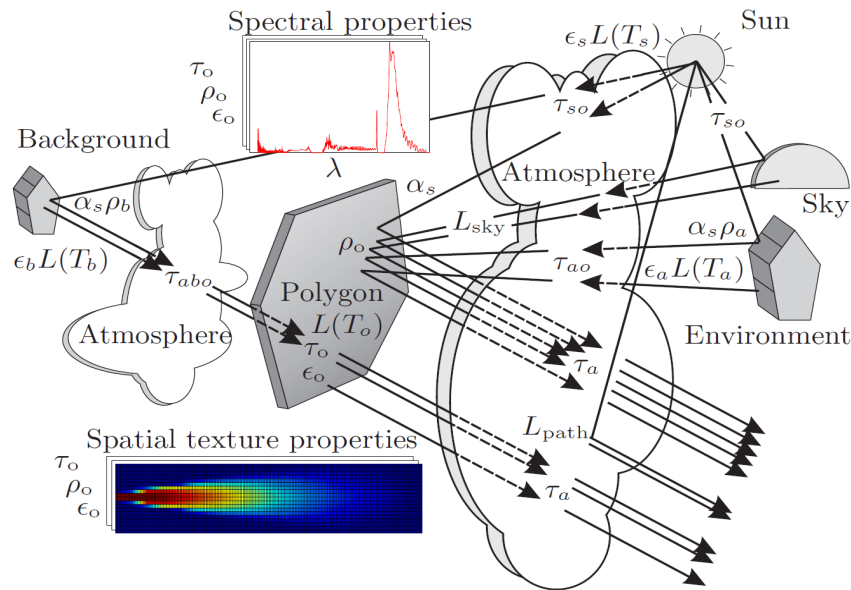


Figure 2.22: OSSIM's radiative transfer equation components [4].

$$\begin{aligned}
L(\Delta\lambda) = & \underbrace{\Delta\epsilon \int_{\lambda} \epsilon_o L(T_o) \tau_a S d\lambda}_{\text{thermally emitted}} + \underbrace{\int_{\lambda} L_{path} S d\lambda}_{\text{atmospheric path radiance}} + \underbrace{\Delta\tau \int_{\lambda} \epsilon_b L(T_b) \tau_{abo} \tau_o \tau_a S d\lambda}_{\text{transmitted background}} + \\
& \underbrace{\Delta\rho \left[\int_{\lambda} \int_{env} \epsilon_a L(T_a) \tau_{ao} \rho_o \tau_a S d\Omega d\lambda \right]}_{\text{reflected background}} + \underbrace{\cos\theta_a \int_{\lambda} \int_{sky} L_{sky} \rho_o \tau_a S d\Omega d\lambda}_{\text{reflected sky}} + \\
& \underbrace{\alpha_s \cos\theta_s \int_{\lambda} \epsilon_s L(T_s) \tau_{so} \rho_o \tau_a S d\lambda}_{\text{diffuse reflected sunshine}} + \underbrace{\left[\frac{\rho_s (n+1) \cos^n \alpha}{2\pi \cos\theta_i} \right] \int_{\lambda} \epsilon_s L(T_s) \tau_{so} \tau_a(S) d\lambda}_{\text{specular reflected sunshine}}
\end{aligned} \tag{2.103}$$

primitives are also assigned a temperature which is calculated with heat balance equation. Terrain thematic maps are constructed by photo-interpretation of remote sensing images, which identify material types used for temperature measurement. OSSIM does not account for discrepancies that may cause due to presence of optical and thermal shadows. Although OSSIM has a comprehensive RT equation, however it does not account for sky-view effect which is an inherent part of DIRSIG equation. Components of OSSIM and their interaction is shown in Figure 2.22.

Table 2.4: Symbols in Equation 2.102 and their meaning, λ shows that quantities are function of wavelength.

Symbol	Meaning	Symbol	Meaning
$L(\Delta\lambda)$	Total radiance in wavelength band $\Delta\lambda$	n	Specular exponent
$L(T_s)$	Spectral black body radiance, sun temperature T_s	τ_o	Object surface spectral transmittance
$L(T_a)$	Spectral black body radiance, environment temperature T_a	τ_a	Object to surface spectral atmospheric transmittance
$L(T_b)$	Spectral black body radiance, background temperature T_b	τ_{abo}	Background to object spectral atmospheric transmittance
$L(T_o)$	Spectral black body radiance, object temperature T_o	τ_{ao}	Ambient to object spectral atmospheric transmittance
L_{path}	Atmospheric path radiance, emitted and scattered	τ_{so}	Sun to object spectral atmospheric transmittance
L_{sky}	Sky radiance, emitted and scattered	Δ_r	Spatial texture variation in transmittance
ϵ_s	Solar surface's spectral emissivity	α_s	$A_{sun}/(d_{sun}^2 \pi) = 2.17 \times 10^{-5}$
ϵ_a	Ambient environment's spectral emissivity	A_{sun}	Area of sun
ϵ_b	Background spectral emissivity	d_{sun}	Distance to sun
ϵ_o	Object's surface spectral emissivity	θ_a	Angle between surface normal and vertical
ρ_o	Object surface diffuse reflectance	θ_s	Angle between surface normal and solar incidence
ρ_s	Object wideband Fresnel reflection	θ_r	Angle between reflected sunlight ray and the viewing direction
$\Delta\rho$	Spatial texture variation in reflectivity	S	Camera spectral response
$\Delta\epsilon$	Spatial texture variation in emissivity		

2.8.5.3 *CameoSim*

CameoSim is one of the earliest and comprehensively feature-rich commercial HSI simulator. Due to a large number of features, it tends to require effort from the end-user before all its features could be exploited. CameoSim is developed in a modular fashion therefore features are divided into modules' user interfaces, typically called editors. Figure 2.23 shows components of CameoSim and data flow.

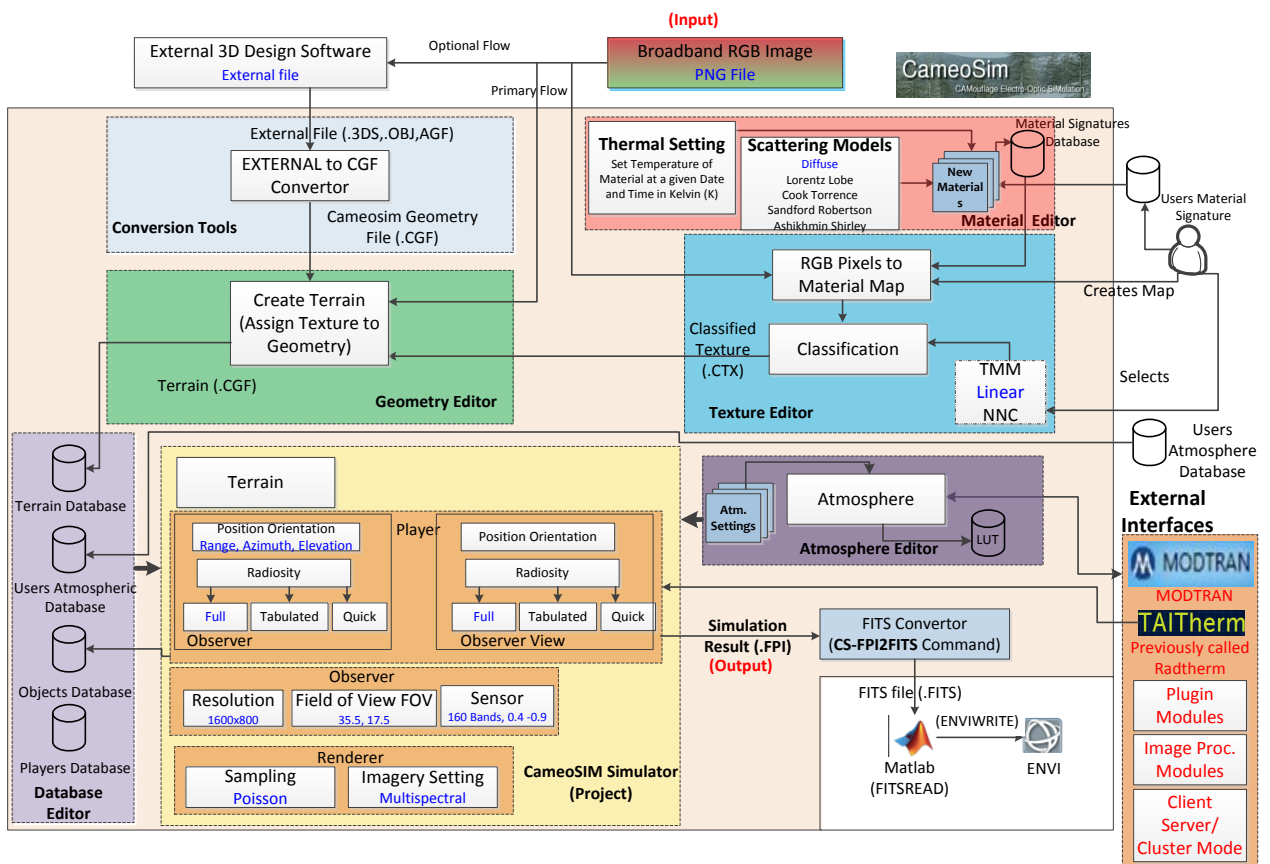


Figure 2.23: CameoSim data flow.

In the process of generating radiometrically accurate synthetic images of scenes. All sources in sythetic environment affect the solution of a rendering equation. The rendering equation for a general radiosity model is given in Equations (2.104-2.107). The three main components of the rendering equation are the thermal self-emission, the atmospheric terms, global illumination accounting for reflected radiance. In CameoSim a general bidi-

rectional, spectral, recursive solution is sought using importance driven Monte-Carlo sampling. At longer wavelengths such as mid-infrared, the full hemispherical integration of incident irradiances enable the software to account for the radiative interaction between different surfaces at varying themodynamical states. BRDF models take account of bi-directional light reflecting properties of a surface, in CameoSim it is termed as (surface) scatter models, as shown in Figure 2.23 [18].

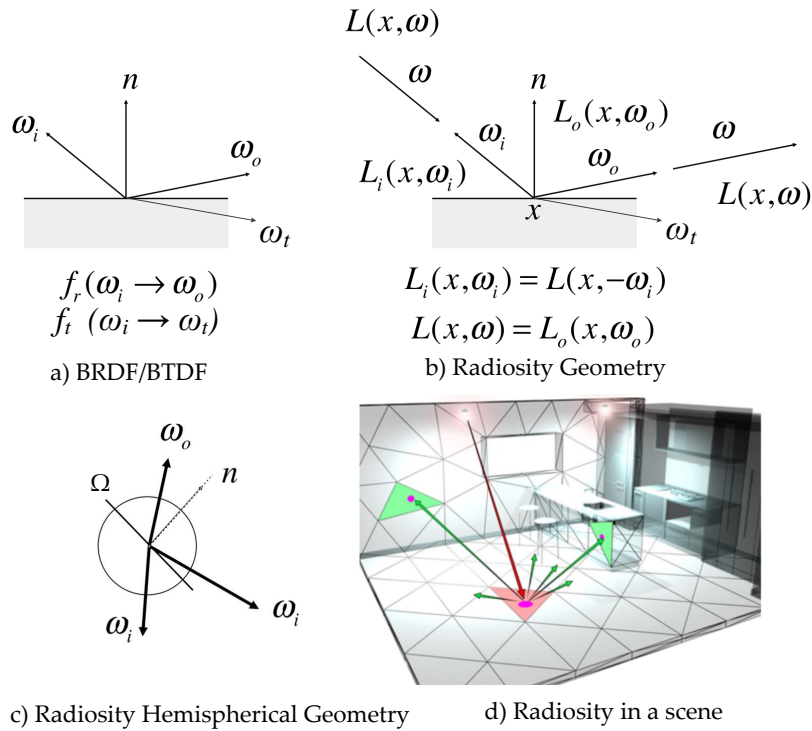


Figure 2.24: CameoSim-like recursive ray-tracing.

Figure 2.24 illustrates components of radiosity rendering equation. Bidirection Reflection Distribution Function (BRDF) and Bidirectional Transmission Distribution Function (BTDF) represent the surface response on interaction with rays. Geometrical definition of sources is shown in Figure 2.24 (b) and the hemispherical integration is included in Figure 2.24 (c). Equations 2.104-2.107 shows the rendering equation for CameoSim. Symbols and their meaning are explained in Table 2.5.

$$L_o(x, \omega_o) = L_{path}(x_1, x_2) + L_e(x, \omega_o) + L_r(x, \omega_o) + L_t(x, \omega_o) \quad (2.104)$$

$$L_r(x, \omega_o) = \int_{\Omega} f_r(x, \omega_i \rightarrow \omega_o) L_i(x, \omega_i) \cos \theta_i d\omega_i \quad (2.105)$$

$$L_t(x, \omega_o) = \int_{\Omega} f_t(x, \omega_t \rightarrow \omega_o) L_i(x, \omega_t) \cos \theta_t d\omega_t \quad (2.106)$$

$$L_o(x, \omega_o) = L_{path}(x_1, x_2) + L_e(x, \omega_o) + \int_{\Omega} f_r(x, \omega_i \rightarrow \omega_o) L_i(x, \omega_i) \cos \theta_i d\omega_i + \int_{\Omega} f_t(x, \omega_t \rightarrow \omega_o) L_i(x, \omega_t) \cos \theta_t d\omega_t \quad (2.107)$$

Table 2.5: Symbols in Equation (2.104-2.107) and their meaning.

<i>Symbol</i>	<i>Meaning</i>	<i>Symbol</i>	<i>Meaning</i>
ω_o	Observer or sensor direction or position	$L(\omega_o, \omega_t)$	Radiance incident on x from direction ω_t
x	A point on the target surface	$f_r(\omega_o, x, \omega_i)$	Bidirectional reflectivity at point x with respect to directions ω_o and ω_i .
ω_t	Incident direction of transmitted radiance w.r.t x	$f_t(\omega_o, x, \omega_t)$	Bidirectional transmissivity at point x with respect to directions ω_o and ω_t
ω_i	Incident direction of emitter's radiance w.r.t. x		
$L_{path}(\omega_o, x)$	Radiance along target-sensor path		
$L_{emit}(\omega_o, x)$	Radiance emitted from point x , towards direction ω_o		
$L(\omega_o, x)$	Radiance reflected from x in the direction of ω_o		
$L(\omega_o, \omega_t)$	Radiance incident on x from direction ω_t		

2.9 Adjacency Effect Models

Adjacency effect is an effect of large background on the observed radiance of a target due to atmospheric back-scattering. It depends on the contrast between a target pixel and relatively larger neighborhood reflectance. In a simulation environment it is importance to model the Adjacency effect to generate realistic HSI images. Due to its dependence on scattering, the adjacency effect is a function of distance between target object and the sensor. It is nominal in distances (< 2 m) and gets substantial in case of airborne (> 1 Km) or satellite imaging (> 100 km). Moreover, like scattering its effect also reduces with increasing wavelength.

To ascertain the true reflectance of a target from a sensor adjacency effect should be compensated. It is therefore a research interest since late 1970s and there are many publications related to correct modelling of it [51, 52, 53, 54]. Figure 2.25 shows

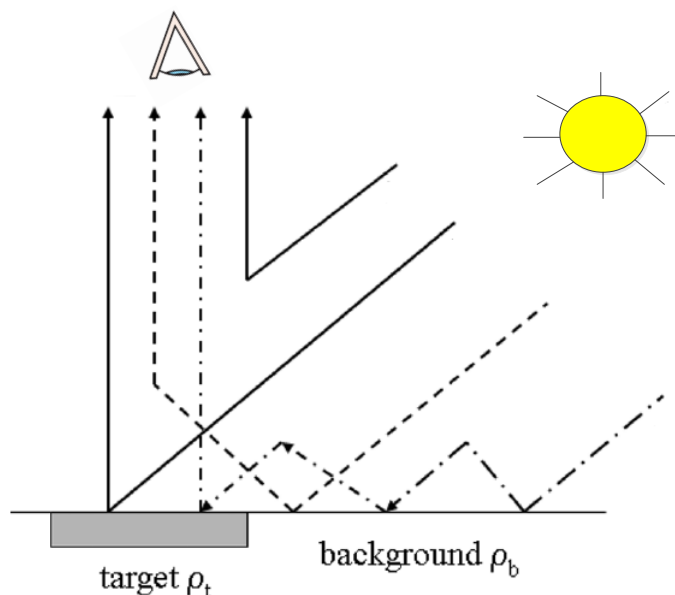


Figure 2.25: Adjacency components shown as dashed lines.

two types of reflected and scattered rays by dashed lines, one is reflected off the background and then scattered into the LoS of the sensor and the second one is due to earth atmospheric coupling which is ultimately reflected from the target towards the sensor. Adjacency model due to [54, 53, 55] are popular and adopted in RT modelling module such as DISORT, these models are described in details in subsequent sections.

Adjacency effect causes discrepancies in index-based classification metrics such as Normalized Difference Vegetation Index (NDVI) and Normalized Difference Water Index (NDWI) etc. Researchers [22, 56] proposed models for adjacency compensation for pixels near waterfront. [22], based on the localization of adjacency effect assumes that all pixels in a scene do not possess adjacency due to waterfront and therefore an adjacency effect can be compensated by comparison of similar reflectance pixels. The one which dominates the water signature is compensation by the one that has a nominal water effect.

Researchers have proposed new adjacency models for rugged terrains [6, 7]. [6] estimated the adjacency effect for Lambertian surface using ray tracing through layers in the atmosphere. In the quantification of adjacency effects, molecular/aerosol scattering phase functions, topographic features and ground heterogeneity are taken into account. This work considers single scattering only to get a good trade-off between computing time and

accuracy. Similar approach is used by [7] but they extended this work

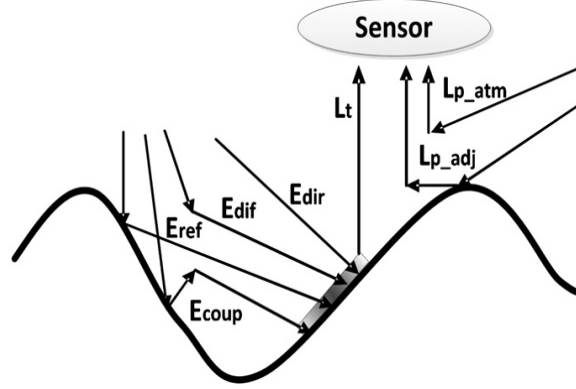


Figure 2.26: Adjacency effect in rugged terrains.[6, 7]

to multiple scattering. Figure 2.26 shows multiple earth atmosphere coupling in adjacency ray before it hits the target and reflected towards the sensor, this phenomena is multiple scattering. This effect is mathematically described in the subsequent section.

2.9.1 MODTRAN

MODTRAN adopts the DISORT algorithm with a variable number of streams for scattering and adjacency modelling. At-sensor apparent reflectance ρ_a is given in Equation (2.108).

$$\rho_a = \frac{\pi L_s}{E'_s \cos \theta}. \quad (2.108)$$

The upward surface flux that is reflected by the atmosphere back to the ground and then reflected by the ground $\bar{\rho}_b$ upwards. Since this phenomena occur multiple times before it is finally scattered into the LoS of the sensor, therefore it takes the form of power series as given in Equation (2.109).

$$(\tau_1^{dir} + \tau_1^{dif})(\tau_2^{dir} \rho_t + \tau_2^{dif} \bar{\rho}_b) [1 + s\bar{\rho}_b + (s\bar{\rho}_b)^2 + \dots] = \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dir} \rho_t}{1 - s\bar{\rho}_b} + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif} \bar{\rho}_b}{1 - s\bar{\rho}_b}. \quad (2.109)$$

At-sensor apparant reflectance ρ_a can be redefined in terms of Equation (2.109) as Equation (2.110) [57].

$$\rho_a = \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dir}\rho_t}{1 - s\bar{\rho}_b} + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif}\bar{\rho}_b}{1 - s\bar{\rho}_b}, \quad (2.110)$$

where ρ_o is the path reflectance. In order to calculate the total upwelled scattered reflectance term, ρ_t should be set to zero, which is target albedo during MODTRAN run. Therefore Equation (2.110) takes the form of Equation (2.111).

$$\rho_u = \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif}\bar{\rho}_b}{1 - s\bar{\rho}_b}, \quad (2.111)$$

where ρ_u is the upwelled scattered apparant reflectance. It includes both path reflectance and adjacent reflectance, in the first and the second term, respectively.

Symbols used in Equations (2.108)–(2.111) are explained in Table 2.6.

Table 2.6: Symbols in Equations (2.109) and (2.110) and their meaning.

Symbol	Meaning	Symbol	Meaning
ρ_o	Path reflectance	τ_2^{dif}	Ground to sensor diffuse transmittance
τ_1^{dir}	Sun to ground direct transmittance	s	spherical scattering albedo of the atmosphere
τ_1^{dif}	Sun to ground diffuse transmittance	ρ_t	Target reflectance
τ_2^{dir}	Ground to sensor direct transmittance	$\bar{\rho}_b$	Area averaged background reflectance

2.9.1.1 CameoSim

Although CameoSim provides surface scattering through built-in BRDF models, it does not include volumetric scattering during the texture rendering. CameoSim treats the adjacency through the DISORT models of the upwelled scattered radiance, which means the adjacency is dependent only on a single source of spectral albedo, that is, the average background albedo of the scene [18].

2.9.1.2 DIRSIGs

DIRSIG5 handles adjacency effect by considering all rays that are scattered into the sensor from outside of its Field Of View (FOV) through the path tracing. This volumetric scattering is driven by the atmospheric phase function of the scene [58], but in this version of the simulator, the scattering efficiency as a function of the sensor altitude has not been

taken into account. Older version of DIRSIG, that is, DIRSIG4 incorporated atmospheric variability [50] and adjacent scattered light can be grouped with the path radiance particularly when the averaged background albedo of the scene is slowly varying [5]. Hence the adjacency in DIRSIG4 is functioning like that of the CameoSim, which utilizes a single stream of the background albedo for the DISORT calculation [18].

2.9.1.3 ATCOR

Although ATCOR is not an HSI scene simulator, it features an adjacency model which does not employ rendering for the estimation of the observed at-sensor radiance in its atmospheric correction package [59]. Jianwen et. al. [60] conducted a series of measurement to estimate adjacency effect, they reported that the adjacency effect becomes stronger from visible, near-infrared to shortwave infrared wavelength. In response to this work, Reference [61] presented Equation (2.115) to include the effect of contrast between the target and background pixels in at-sensor radiance. Before deriving ATCOR's model the RT equation can be rephrased in terms of atmospheric spherical albedo for a "large surface" with reflectance ρ , as follows [54, 55],

$$L_s(\rho) = L_p + \frac{\tau E_g(0)\rho/\pi}{1 - s\rho}, \quad (2.112)$$

where τ , $E_g(0)$, and s are the total ground-to-sensor transmittance, global flux on the ground for $\rho = 0$, and the spherical albedo of the atmosphere, respectively. τ is the sum of the direct and diffuse transmittance, that is, $\tau = \tau_{dir} + \tau_{diff}$. Equation (2.112) shows that the effective global flux is;

$$E_g(\rho) = \frac{E_g(0)}{1 - s\rho} \quad (2.113)$$

and it depends on the ground reflectance and spherical albedo [61]. However, for "small target" of reflectance ρ_t , the at-sensor radiance is calculated as given in reference [55].

$$L_s(\rho) = L_p + \frac{\tau_{dir} E_g(0)\rho_t/\pi}{1 - s\bar{\rho}_b} + \frac{\tau_{dif} E_g(0)\bar{\rho}_b/\pi}{1 - s\bar{\rho}_b}. \quad (2.114)$$

Furthermore, Richter et. al. [61] rewrote Equation (2.114) to accommodate the contrast term;

$$L_s(\rho) = L_p + \frac{\tau E_g(0)\rho_t/\pi}{1 - s\bar{\rho}_b} + \frac{\tau_{dif} E_g(0)(\bar{\rho}_b - \rho_t)/\pi}{1 - s\bar{\rho}_b}. \quad (2.115)$$

Equation (2.115) emphasizes the fact that the adjacency term is directly proportional to the target background reflectance contrast. ATCOR also includes sensor height based scattering efficiency calculation based on air density, and volume between the sensor and the target material [18].

2.10 Path Tracing vs Ray Tracing

In ray tracing, there are three types of rays i.e reflected, refracted, and shadow rays. When a ray interacts with an obscured surface, a ray continues its specular reflected direction when it hits a shiny surface.

Refracted rays propagate through the transparent material which may either reflect at the boundary of the surface due to total internal reflection (caused by a certain incident angle) or otherwise exit the material. Although it is a recursive process, however, to avoid tracing all rays in a scene, a shadow ray is used to test if a surface is visible to the light. If this test is true then a ray is traced between this intersection point and the light.

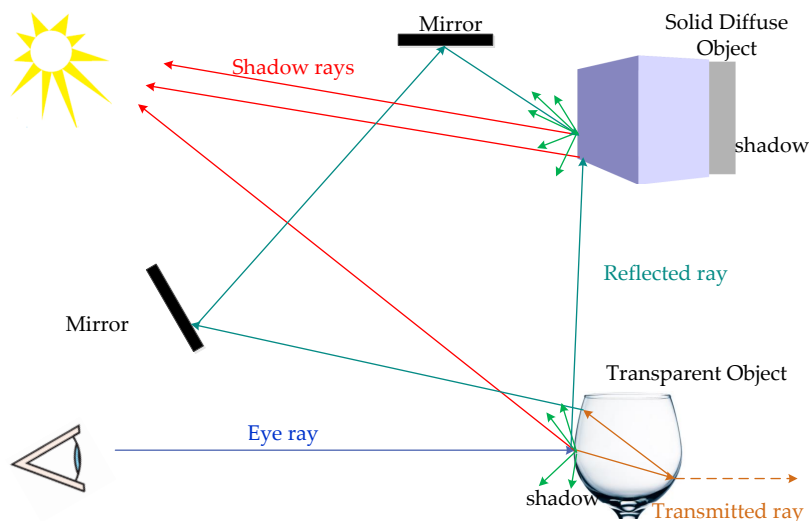
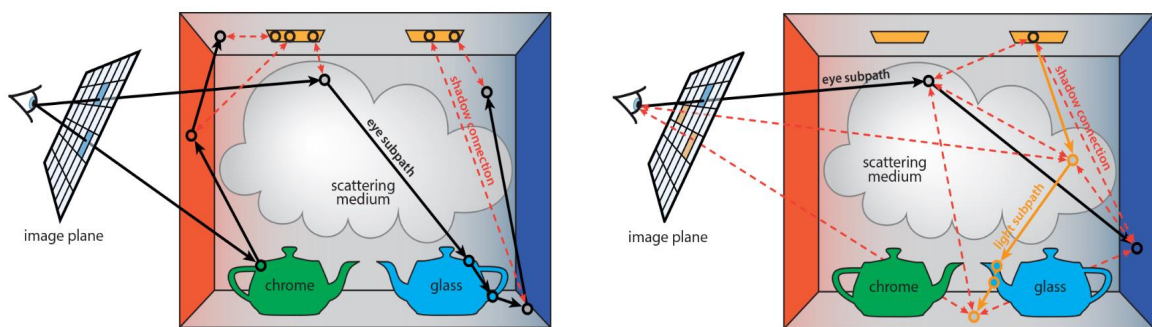


Figure 2.27: Ray tracing setup, at the diffuse and transparent object intersection the reflected rays (green) are spawned into new similar rays. In transparent objects transmitted rays (orange) pass through the object. When a shadow is cast due to occlusion towards a background region, shadow rays (red) are cast in the opposite direction of the shadow, towards the emitter. An eye ray (blue) is cast from the eye or sensor.

In the case of path tracing, a path history of a single point is created when a ray

interacts with the environment until it is fully absorbed. That is, rather than to spawn new rays at the intersection point, it simply chooses a direction as according to the BRDF of the surface and to assign the direction for the ray to follow. A number of random rays per pixel are then sampled using the Monte-Carlo method and the final at-sensor radiance is computed through image reconstruction filter, which rasterizes the accumulated radiance. An important property of path tracing is that the computation is more emphasis on those low-depth rays, while the Whitted ray tracing concerns more of the high-depth rays.



(a) Path tracing from observer to scene and scene to emitter. (b) Bidirectional path tracing, from observer to scene and emitter to scene.

Figure 2.28: Path tracing.[8]

There are two popular versions of path tracing, one is unidirectional from the observer (image plane) to the scene and to the emitter. Another bidirectional approach due to [49] where a path from emitter and observer are traced towards the scene [18].

2.10.1 Rendering Schemes

In section 2.7 we discussed the individual terms and components of accumulated observed radiance. Here we will discuss integral of those terms during path tracing.

2.10.1.1 Brute Force Ray Marching

In all the terms we see nested integrals one for computing the transmittance and the other for computing the incident solar irradiance. One early proposal to address this issue was raymarching algorithm [62] in which traversing along the ray in fixed steps of ray length, both transmission and irradiance are computed. However, in order to compute

the shadowing the ray marching has to be performed recursively, therefore, this type of brute-force setup takes $O(N^2)$ where N is the number of steps of ray lengths.

2.10.1.2 Stochastic

Bi-directional path tracing in which rays are traced from the sensor (camera) and the light source is a type of stochastic approach. This technique is also known as distance sampling, as it traces rays randomly and computes the scattered radiance along the path at every scattering event.

Suppose we select a point along the ray in $O(N)$ then we need another $O(N)$ steps again to compute shadow rays to the light source, so the algorithm is now linear in the number of steps, but the resulting rendered image tends to get noisy. Furthermore, more transmittance and irradiance are still coupled as it is not possible to increase the quality of one independently.

An acceleration based on data structures such as kd-trees has been proposed to determine an upper bound on extinction coefficient per ray, allowing optically thin regions to take fewer steps. Kd-trees divide the scene based on primitives. First partitioning criteria (dimension) and then partitioning based on the presence of primitives in a particular plane is performed. The partitioned scene coordinates are then stored in the kd-tree which has $O(\log N)$ where N is the number of scene partitions.

2.10.2 Sampling Methods

- Hammersley sampling

The two-dimensional Hammersley point set of order m is defined by taking all numbers in the range from 0 to 2^{m-1} and interpreting them as binary fractions. Calling these numbers x_i , then the corresponding y_i are obtained by reversing the binary digits of x_i . For example, for the Hammersley point set of order 2 are given by 0.00, 0.10, 0.01 and 0.11 or (0, 1/2, 1/4, 3/4). Reversing the bits then gives the second component, leading to the set of points (0, 0), (1/2, 1/4), (1/4, 1/2), and (3/4, 3/4). The Hammersley sampler obtains a very high-quality point set that is slightly more regular, the first few dimensions. An example of low discrepancy Hammersley sampling is shown in Figure 2.29.

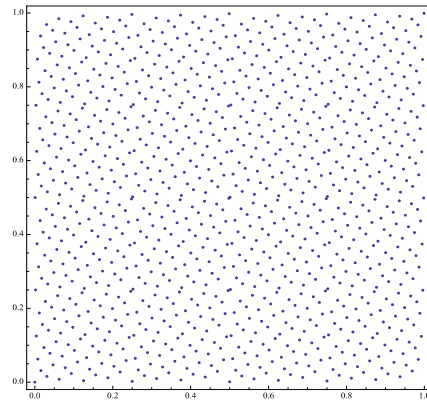
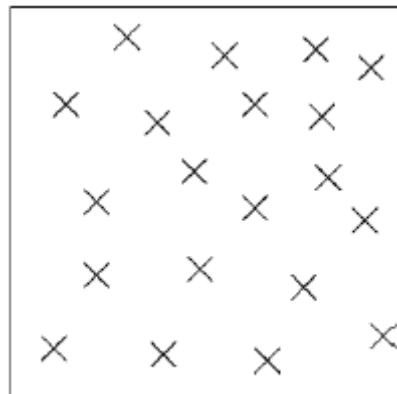


Figure 2.29: Hammersley sampling in a unit area

- Poisson sampling

In this mode, several rays are fired as clump at random regions within the pixel, and the average is taken as a sample random point. In choosing the next random points, the renderer tries to choose an even distribution so points are not too close to each other. An example of Poisson sampling is shown in Figure 2.30.



Poisson

Figure 2.30: Poisson sampling in a unit area [9]

2.10.2.1 Image Reconstruction

Once the integration process collects all the rays after path tracing, an image is constructed. Texture aliasing is the distortion or artefact that results when a signal reconstructed from samples is different from the original continuous signal. Secondly, the

quantification of intensities in pixels from the ray radiance should also be estimated. In order to have a good visual quality reconstruction both these challenges should be addressed.

Textures are maintained in the memory during rendered in the form of pyramidal layers of downsampled bitmaps, which is also referred to as mipmap. One approach to performing anti-aliasing on texture is called Elliptically Weighted Average. The EWA filter performs anisotropically filtered lookups on two adjacent mipmap levels and blends them. This produces the best quality but at the expense of computation time.

In order to reconstruct the pixel intensity a **gaussian windowed filter** based averaging is a simple and natural form of reconstruction filter and it is also adopted in the simulator.

Chapter 3

CHIMES Simulator

In this chapter we introduce the features and research on CHIMES simulator. It is however, important to cover a few basic concepts regarding Hyperspectral Scene Simulation. These details are discussed in this chapter.

3.1 Proposed CHIMES Simulator

CHIMES is a texture rendering based simulator and an overview of its design and system components are shown in Figure 2.6 and Figure 3.1 respectively. As mentioned in the introduction section, there are two main features in the proposed CHIMES; one is enhanced adjacency treatment that has been implemented for a better estimation of the volumetric scattering of upwelled scattered radiance, which is particularly important for the simulation of rugged scenes. This feature is in the form of a regional background neighbourhood function, which, embeds an altitude dependence of the optical scattering characteristics between the ground and the sensor. During the simulation, the averaged background albedo of the surrounding region of the test pixel, and the total upwelled radiance are evaluated for each pixel in the scene during the rendering process. The use of this neighbourhood function together with volumetric scattering has not been found in the open domain, and it is believed that this approach may help to enhance the performance of other scene simulators in the community. DIRSIG5 offers volumetric scattering based adjacency model, however considering effective background neighbourhood based on sensor height, air volume and density is not reported. Although ATCOR also employs the neighbourhood based adjacency, however, it performs differencing of target and background spectra to model the reduced contrast. This is only an approximation because volumetric scattering is not considered in ATCOR, probably due to its application domain. Moreover, variable height difference from every pixel to the sensor is not also

considered in ATCOR or any other COTS scene simulation package.

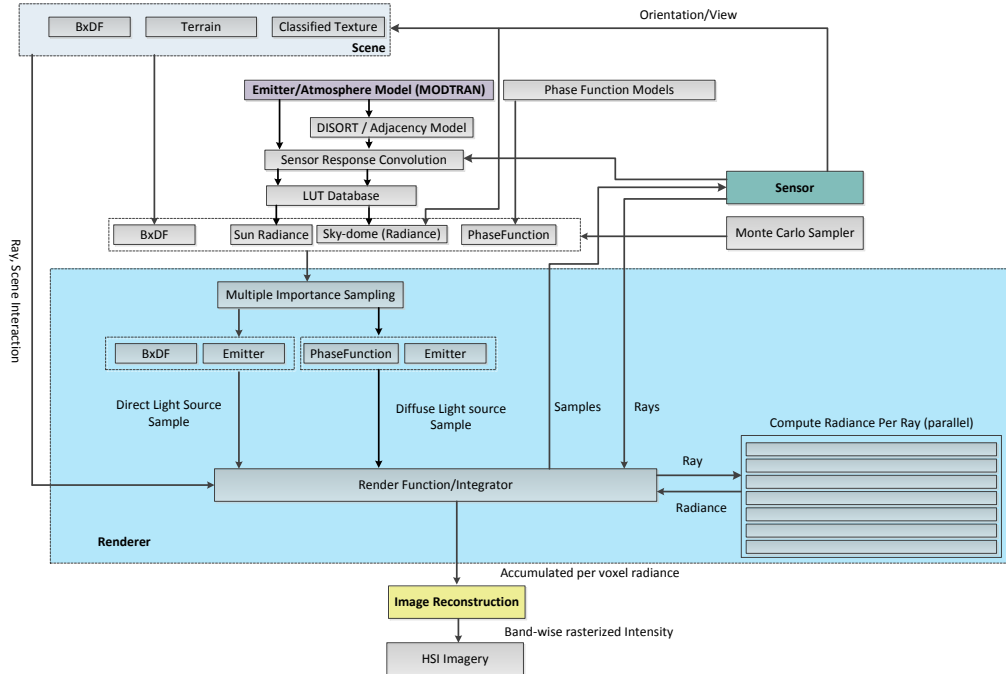


Figure 3.1: CHIMES Simulator, system level design overview. Sensor, Scene, BRDF, Renderer and Image Reconstruction modules are shown in different components as coloured boxes.

The other feature in the CHIMES is the automatic search of the optimal atmospheric parameters to be used for the simulation of a scene, such that the output will be as realistic as that of the ground truth data. Many simulators, such as the DIRSIG and the CameoSim, require the user to define the atmospheric parameters which are unknown. Thus several trial and error of simulation run using different sets of atmospheric parameters are normally required for these simulators. This contribution will help to reduce the amount of guesswork and help to obtain the optimal simulation result without repeatedly trial and error processing by varying the aerosol optical thickness. Contents of this chapter are based on [18].

3.1.1 CHIMES Data-Flow

Figure 3.2 shows an overview of the software framework of CHIMES simulator. It is an evolvable software framework which allows users to add custom modules related to the

simulation and provide integration of other built-in simulation components such as 3D geometry, Atmospheric data, Sensor Data, Material signature database, a Graphic User Interface and a rendering engine. In order to make the software design more modular and we have divided it into four layers such as;

- Graphic User Interface
- Scripting Language Layer
- Middle Logic Layer
- Database Interface

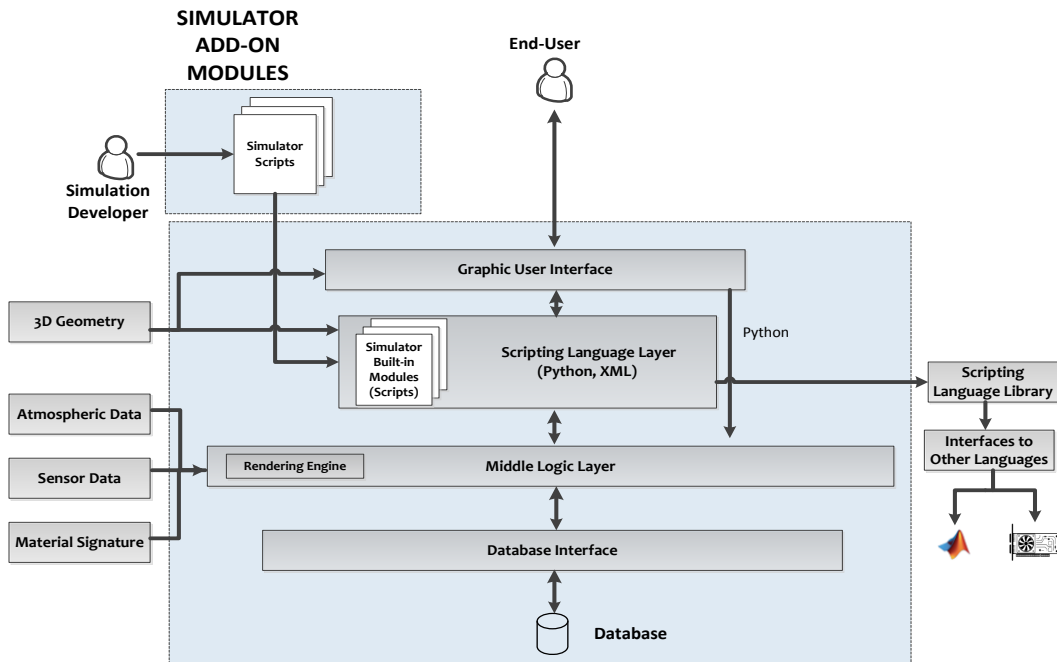


Figure 3.2: CHIMES components, data flow and user interaction methods

The objective of this design is to allow users to develop the simulator with scripts. Script registration can be written to define the scene parameters and start the rendering process. Screenshot of CHIMES simple GUI is shown in Figure 3.3.

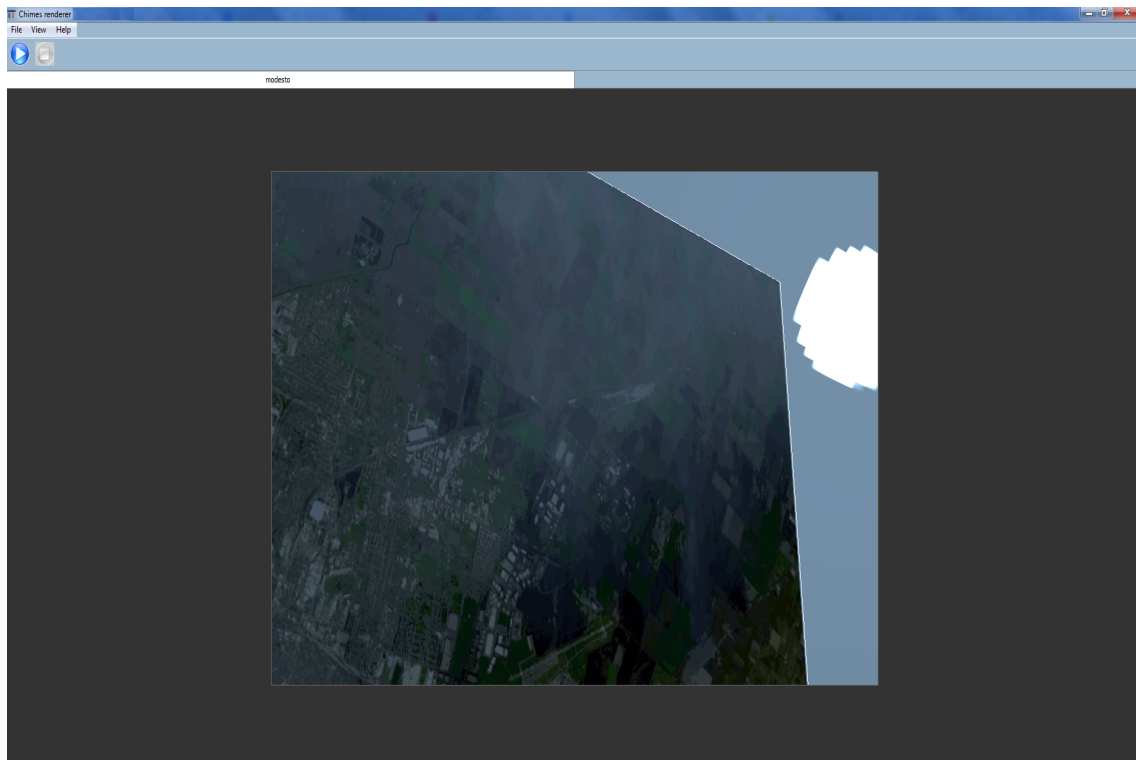


Figure 3.3: CHIMES Simulator simple GUI, showing a pre-rendering RGB view of Modesto Scene, partly showing sky and sun. Scene can be defined as python script or XML definitions file.

3.2 Components of CHIMES

3.2.1 Scene

The terrain is generated by DEM-based smoothed deformation in the flat polygon, which is the basic primitive of a scene. The input data is airborne reflectance images with the high spatial and spectral resolution, which is processed by QUAC atmospheric compensation on radiance image. Classification of reflectance image cube is the first step for material mapping, which is performed by K-means clustering algorithm with a user-defined number of materials (classes) input. Details about classified texture are covered in a subsequent section. The number of specified material and their identifiers are created and stored in the database. A texture is generated based on materials RGB colours. An identifier based coded-texture layer is generated to handle the loading of spectral data during rendering. Corresponding BRDF/BTDF is associated with each material, however, the

thermal description is out of the scope of this work. This classified texture is mapped onto the terrain to create static geometry. A potentially roving geometry as shown in Figure 2.6 might be represented by vehicles, plumes or clouds, however, it is not included in this version of the simulator.

In terms of rendering system, a scene is an input to the rendering algorithm which recursively cast rays to test occlusion for casting shadows or perform an interaction with scene surface to determine the ray response according to reflection or transmission distribution functions. Another important interaction of the scene is with the sensor, the orientation, altitude and azimuth of the sensor create a perception of the scene on the sensor. Although the skydome itself is not part of scene geometry, however, it may include substantial space in an outdoor scene, viewed by the sensor.

3.2.1.1 Classified Texture

CHIMES is capable of simulating and rendering large high resolution HSI images. For example, one of the scenes rendered has a resolution of (1850×380) which is approximately 700,000 pixels in one image. This equivalently implies that this scene has 700,000 materials. Storing information about this large number of materials would be a waste of memory and would also lag down the rendering process. It is therefore wise to perform classification of the texture so that the number of materials required to represent an image would reduce to process the rendering efficient under a decent memory requirement. A number of classification algorithms such as Support Vector Machines (SVM), K-means clustering, ISODATA were test in pre-selection phase on various HSI datasets.

In our experiments, K-means clustering is employed for classification due to better performance among the alternatives, which is performed on an atmospherically compensated (by QUAC [63]) reflectance hyperspectral cube. In this phase of research where the focus is on RT models, classification from MSI or RGB image like CameoSim/DIRSIG is out of the scope. K-means clustering requires a user-defined number of classes as input. RGB image of classification from 80 and 120 classes are shown in Figure 3.4 (a) and (b), respectively. These classification results are performed in Matlab GPU based implementation using 100 K iterations.



(a) 80 classes, orange perspex appears to have white colour



(b) 120 classes, orange perspex appears to have orange colour.

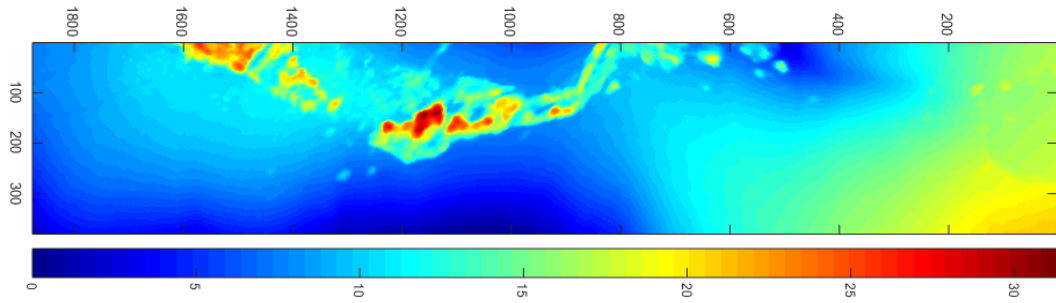
Figure 3.4: Reflectance cube retrieved by QUAC, K-means clustering is utilized for classifying the reflectance. Classified textures with 80, 120 K-means classes are shown in (a,b).

A flat polygon is transformed into the terrain by introducing DEM driven realistic bumps onto it. DEM is measured in Absolute Ground Level (AGL) for the test scene and it is shown in Figure 3.5 (a). The classified image from Figure 3.4 (b) is mapped on the terrain to construct a classified texture, as shown in Figure 3.5 (b).

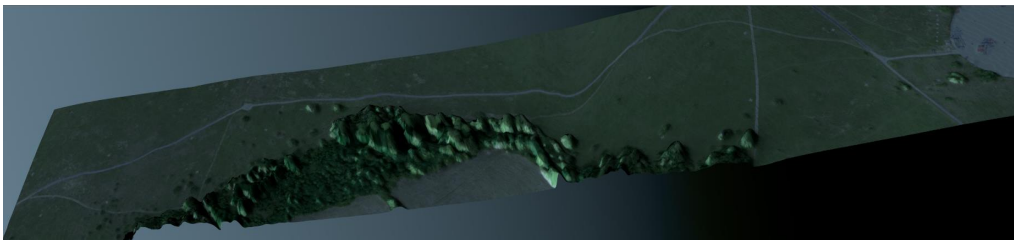
3.2.2 Sensor

Radiance generated by MODTRAN has a very high spectral resolution ranging from wavenumbers ($\tilde{\nu}$) between 3800 to 28000 (cm^{-1}). To simulate a real sensor the output is spectrally degraded to represent the instrument characteristic spectral channels. This is achieved through convolution with sensor response functions. The normalised convolved radiance values in band i are calculated as;

$$L_{si} = \frac{\int_{\lambda} L_{mi}(\lambda) r_i(\lambda) d\lambda}{\int_{\lambda} r_i(\lambda) d\lambda}, \quad (3.1)$$



(a) Digital Elevation Model (AGL) of the experiment scene. Height is given in meters, the highest point is tree top that is 31.3 m with respect to the trough which is 0.0 m.



(b) Classified texture

Figure 3.5: Terrain created from DEM and background skydome in CHIMES software (an RGB view).

where L_{si} is convolved radiance of i th band, L_{mi} is MODTRAN high spectral resolution radiance and r_i is the response function in i th band [64]. The sensor module also takes part in the rendering system by accepting the samples and returning rays distributed on the detector's view grid, whose count is defined by the user as rays per pixel. A scene is visible to the view grid by depth-sensitive perspective projections.

3.3 Atmosphere and Emitter Construction

In this section we will discuss the implementation aspect of how the atmosphere is generated in a simulation environment from the output of MODTRAN, which gets a number of **specific** geometric and altitude parameters to produce output radiance. In order to create a hemispherical atmospheric effect into a simulator, MODTRAN is run several times to cover the directions of the spherical coordinates. Components of at-sensor radiance are generated by means MODTRAN-based atmosphere. Each component is illustrated in subsequent sections.

3.3.1 Direct Solar

The solar incident light is shown to be incident on the ground and depends on the single solar zenith incident angle with respect to target surface normal. Figure 3.6 shows data points for which MODTRAN data is extracted for the solar downwelling radiance.

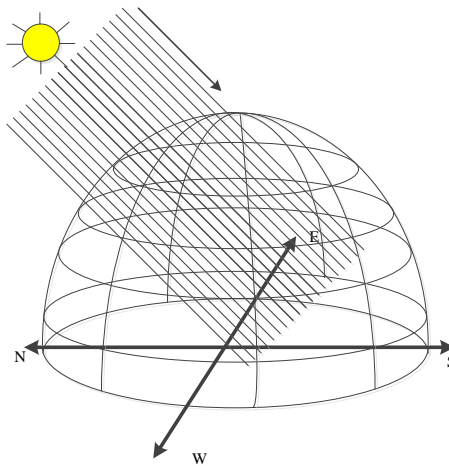


Figure 3.6: Downwelled solar irradiance

Computing the incident flux is, therefore, a straight forward process which is a function of exoatmospheric solar irradiance, transmittance along the LoS and solar zenith angle as expressed in Equation (2.45). In the emitter, environment map sun is placed according to its zenith and azimuth angles.

3.3.2 Sky radiance

In order to construct an environment map in the simulator, the total sky irradiance is incorporated in the environment map which is constructed by computing sky radiance at data points all over the hemisphere. Figure 3.7 shows incident solar irradiance onto the hemisphere, while Figure 3.8 shows data points for which sky irradiance MODTRAN data is extracted for the given atmospheric configuration. The environment map in CHIMES is a (1024x512) hyperspectral image, transformed into spherical coordinates. As the image resolution is higher than the number of time MODTRAN data is acquired therefore there are coordinate points of the atmosphere where interpolation is performed. The face of the

hemisphere where interpolation is to be performed is shown in Figure 3.8.

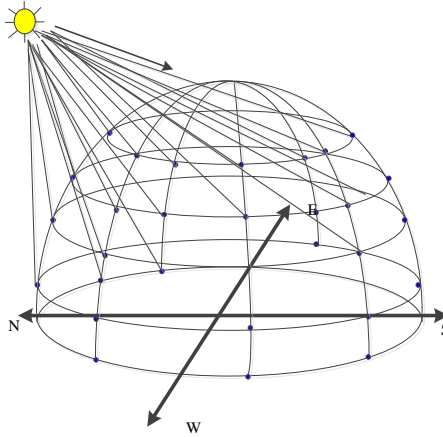


Figure 3.7: Solar irradiance in it's first leg is incident at the data points on the hemispherical horizon of the scene. Data point are selected spherical coordinates where MODTRAN's scattering, transmission etc are computed.

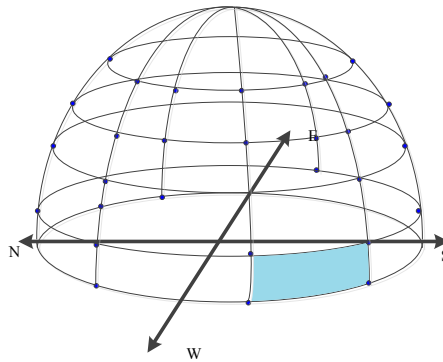


Figure 3.8: Data points on the hemispherical coordinates that are scattering points for downwelled radiance. The blue face shows the plane where interpolation is required to construct the sky environment map.

To construct the database of radiance, transmission and scattering coefficients a composite key is constructed combining zenith and azimuth angles of the data points. MODTRAN is then run based on different altitude and these geometrical angles for a given configuration as shown in Table 3.1 or other relevant settings for a given scene. All atmospheric data is stored in the database and loaded into memory during rendering for

creating the skydome.

3.3.3 Upwelled Radiance

Once the sky and sun emitters are constructed it creates the incident light rays on the scene. Light reflected off the target surface is scattered into the LoS of the sensor. Figure 3.9 shows the Field of View (FoV) of the sensor. The upwelled radiance comprises all components of the at-sensor radiance that are given in Equation (2.61). The sun and sky radiance are reflected according to the BRDF of the target surface however path radiance is calculated from MODTRAN for several geometrical data points as shown in Figure 3.9.

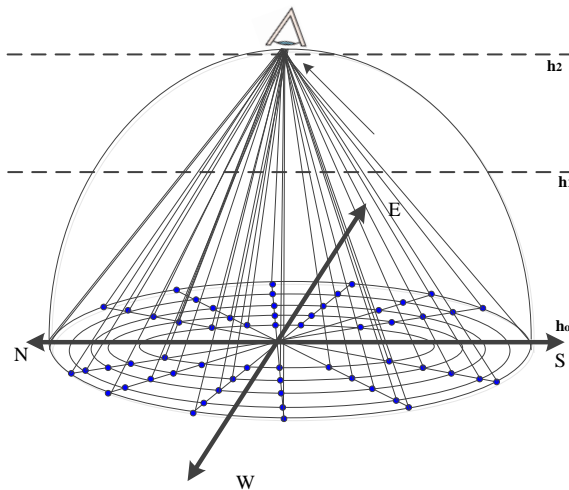


Figure 3.9: Upwelled scattered radiance with spherical and altitude variability. Blue points show the geometric points where data is acquired from MODTRAN, at heights h_0 , h_1 , h_2 .

MODTRAN data is also acquired at heights h_0 , h_1 , h_2 and transmission at these heights are computed. For each layer, a spectral extinction coefficient is associated which is used to estimate the path radiance with varying heights using exponential interpolation.

3.3.4 Skydome Radiance Map

Although MODTRAN has provided irradiance models for the emitters such as the sun, moon [65], in CHIMES the sun and sky irradiance have been implemented as a skydome which is generated through interpolation of MODTRAN's look-up table (LUT) for various zenith and azimuth settings. MODTRAN is typically run with multiple scattering models (DISORT scaled Isaac) and Mie scattering for the appropriate atmosphere (aerosols) of the scene given by the atmospheric search algorithm as described in Section 3.6. A skydome degree of polarisation (DOP) is achieved by rayleigh scattering on environment maps pixel position (zenith, azimuth) with respect to solar position, polarimetric version of MODTRAN called MODTRAN-P [66] is not used to generate this skydome. The sensor response is convolved with the MODTRAN radiance as described in Section 3.2.2. Figure 3.11 (a), plots the variation of radiance for skydome with clouds, while Figure 3.11 (b) shows the DOP of the sky for depicted (by yellow circle) solar azimuth and zenith positions. It is seen from these figures that a strong polarization of solar irradiance is observed at 90° with respect to solar vector and is caused by the Rayleigh scattering of the air molecules in the atmosphere [67]. It should be noted that clouds are not added as roving geometry in this version of CHIMES, however, reduced radiance due to the presence of cloud is calculated from MODTRAN and skydome with lower radiance is created. A polar plot of skydome radiance in overcast weather is shown in Figure 3.12 (a). A similar plot for clear sky, that is, with no aerosol is shown in Figure 3.12 (b). The DoP in sky due to Rayleigh scattering is given in Equation (3.2) which is due to [67]. The DoP plot is shown in Figure 3.10 [18].

$$DoP = \frac{1 - \cos^2 \theta_v}{1 + \cos^2 \theta_v} \quad (3.2)$$

where θ_v is the scattering angle.

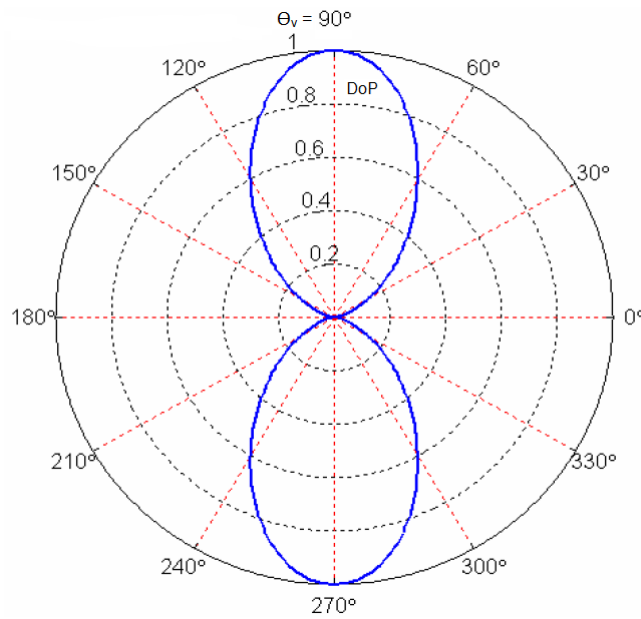
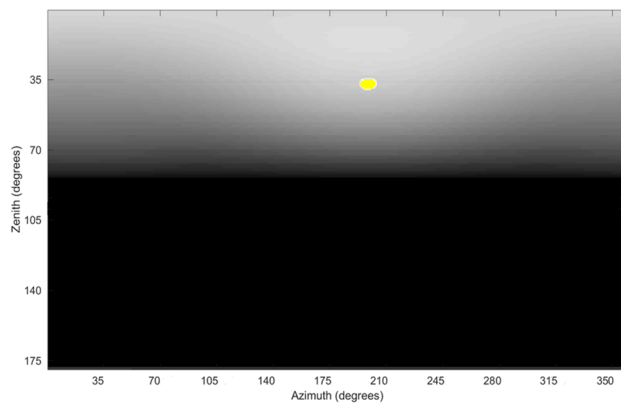
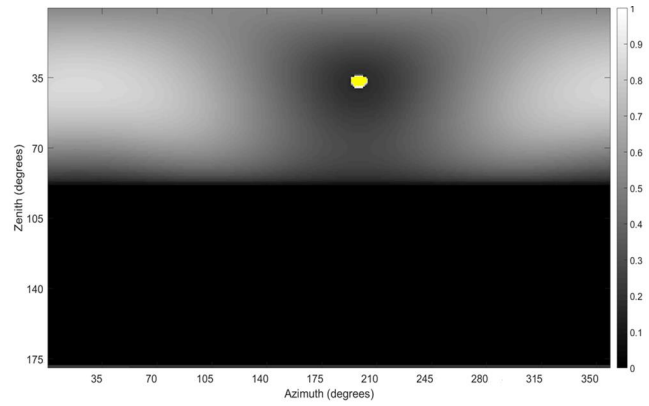


Figure 3.10: Degree of Polarisation with respect to zenith and azimuth



(a) Skydome radiance variability with respect to zenith and azimuth.



(b) Skydome degree of polarisation.

Figure 3.11: CHIMES sky and sun emitters.

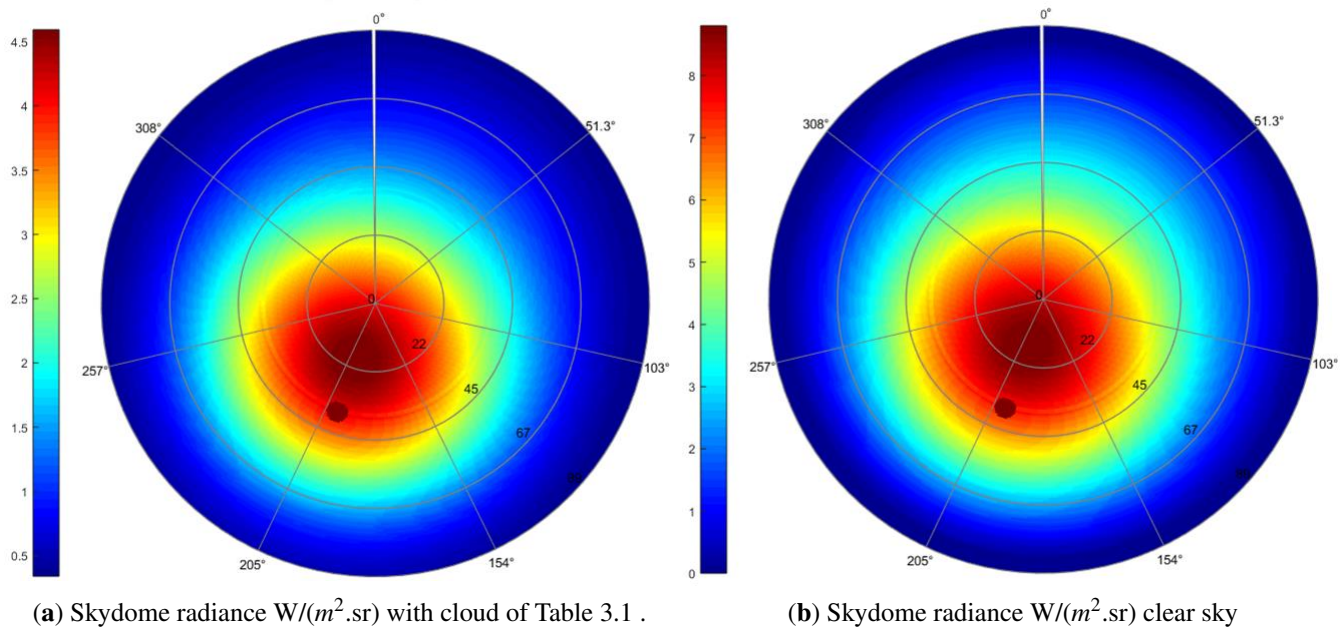


Figure 3.12: CHIMES emitters.

3.4 Spectral Rendering

The primary difference between a visible image and HSI image rendering is in the high spectral resolution rendering. Constructing a classified texture is covered in section 3.2.1.1, the spectral rendering in CHIMES is constructed by making a material identifier texture layer which has a one to one mapping with the image texture. When a ray intersects a scene, texture coordinate of the scene is retrieved for both image and material identifier texture layers. The identifier is used to query the material mapped to the texture coordinate. A bxdf identifier is also recorded against the spectral identifier which attaches a particular BXDF with that particular material. When both this information are accessed bxdf sampling is performed and a vectorized bxdf weight is calculated for that particular texture coordinate. Figures 3.13 and 3.14 show examples of Selene and Modesto material identifier textures images.

Each texture image is a 24-bit colour image, therefore, it can handle up to 16 million materials represented by their material identifiers as shown in Figure 3.15. In CameoSim this is limited to 256 materials for an image.

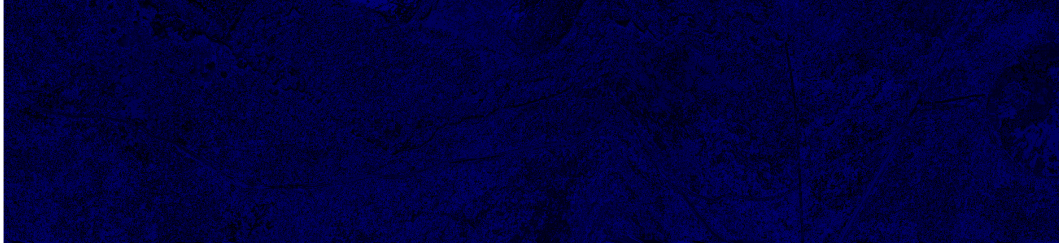


Figure 3.13: Material ID texture of Selene scene.

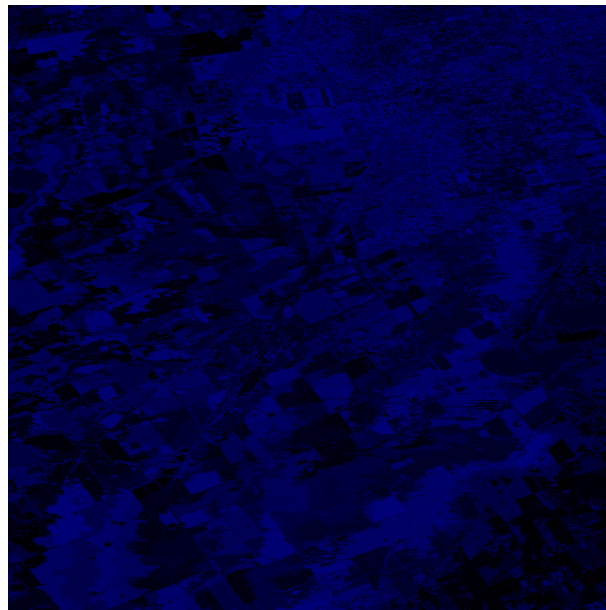


Figure 3.14: Material ID texture of Modesto scene.

Figure 3.15 elaborates the database structure of CHIMES and shows the texture material database table to develop a detailed understanding of how data is handled during the rendering process. The material identifier textures in Figure 3.13 and 3.14 are predominantly blue because there are 120 materials used in our simulations. The number of materials equal to or less than 256 will lie in the blue byte of the texture image therefore it appears blue.

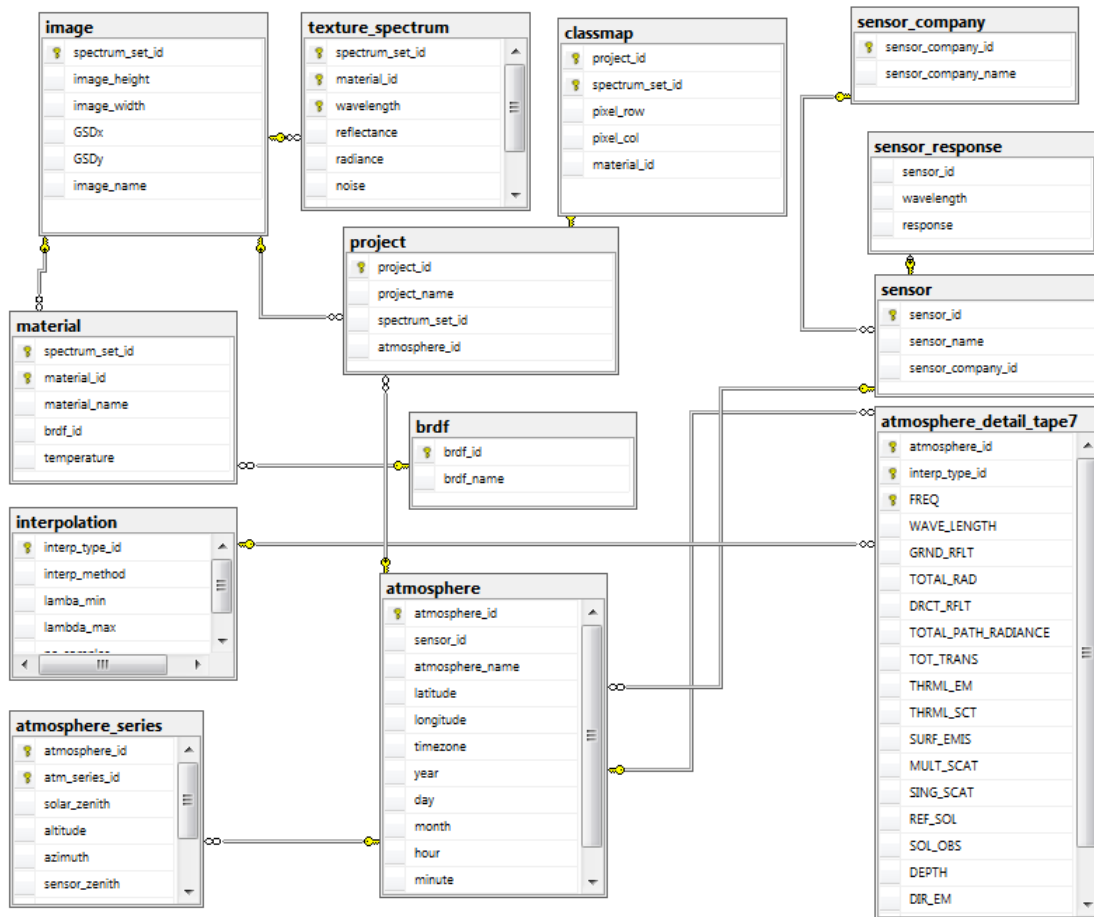


Figure 3.15: CHIMES Database structure

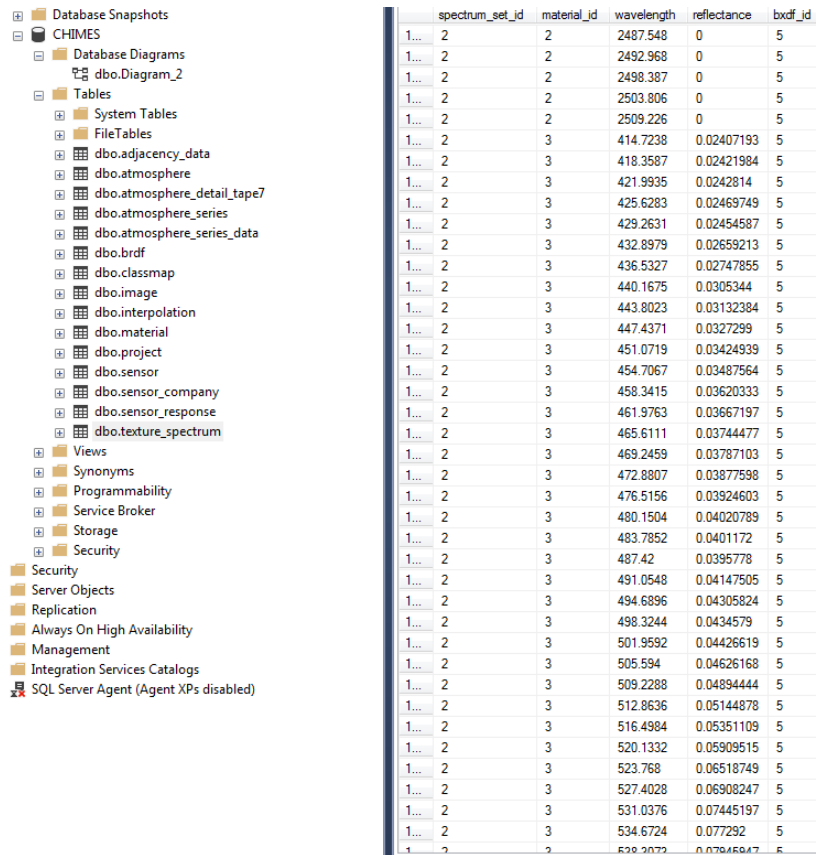


Figure 3.16: Selene texture data.

3.5 Radiometric Integrator

In this section we will discuss the rendering equation employed in CHIMES, flow-chart for computing radiance in the renderer, partitioning and searching of scene regions during the rendering and reconstruction and interpolation methods.

3.5.1 CHIMES Rendering Equation

As we have established a localized form of computing upwelled scattered radiance in the previous section and based on descriptions given in section 2.7 we may now establish a rendering equation estimate based on MIS Monte Carlo integration. Equation (3.3) expresses the rendering equation adopted in CHIMES. In comparison with Equation (2.61), it can be noticed that thermal emission and scattering terms are ignored, as per scope. Figure 3.17 shows an overview flowchart of rendering based on Equation (3.3).

$$\begin{aligned}
L_{\text{sensor}} = & \overbrace{\left\{ \frac{1}{N_f} \sum_{i=1}^{N_f} \frac{f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) L_{\text{sol}}(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) w_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))}{\text{pdf}(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)} \right\}}^{\text{direct solar}} + \\
& \overbrace{\left\{ \frac{1}{N_f} \sum_{i=1}^{N_f} \frac{f(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) L_{\text{sky}}(\theta_i, \phi_i, \theta_r, \phi_r, \lambda) w_f((\theta_i, \phi_i, \theta_r, \phi_r, \lambda))}{\text{pdf}(\theta_i, \phi_i, \theta_r, \phi_r, \lambda)} \right\}}^{\text{diffuse sky}} + \\
& \overbrace{\left\{ \frac{1}{N_{\beta_{op}}} \sum_{j=1}^{N_{\beta_{op}}} \frac{T_p(\theta_{uo}) \beta_{op}(\theta_{uo}) w_g(\theta_{uo})}{\text{pdf}(\theta_{uo})} \right\}}^{\text{path radiance}} E_{TOA} + \\
& \overbrace{\left\{ \frac{1}{N_{L_{uo}}} \sum_{j=1}^{N_{L_{uo}}} \frac{L_{uo}(T, T_{dif}) \bar{\rho}_b(T, T_{dif}) w_f(T, T_{dif})}{\text{pdf}(T, T_{dif})} \right\}}^{\text{adjacent radiance}} + \\
& \overbrace{\left\{ \frac{1}{N_{L_b}} \sum_{j=1}^{N_{L_b}} \frac{L_b(\vec{\omega}_b)}{\text{pdf}_{L_b}(\vec{\omega}_b)} \right\}}^{\text{background reflected radiance}} \quad (3.3)
\end{aligned}$$

Equation (3.3) is a Monte Carlo estimate of at-sensor radiance based on MIS. The direct solar term is already given in Equations (2.87), similarly (2.99) is a componentized version of sky radiance for explanation while we utilize radiance from MODTRAN output therefore we reduce it to L_{sky} . The BRDF term is included as suggested in [49]. The direct and diffuse sky radiance is denoted as L_{sol} and L_{sky} , respectively. The path radiance is incident on to the atmosphere and is scattered into the LoS of the sensor, having transmission T_p . The angle θ_{uo} is upwelled scattering angle, where the subscript o denotes that we are dealing with optical scattering (β_{op}). $N_{\beta_{op}}$ are the samples of functions representing transmission and optical scattering, respectively. In adjacent radiance, L_{uo} is the upwelled scattered radiance which is a function of total downwelling transmission T and diffuse upwelled transmission T_{dif} . According to [53, 61] the adjacency effect is due to diffuse upwelled transmission and accumulates the background reflectance ρ_b as a function of the distance between target position and sensor position. In background radiance term $\vec{\omega}_b$ is the background's solid angle and L_b is the background radiance incident

on the target. Figure 3.17 shows a loop of recursive ray casting and computing of L_{sensor} for each ray.

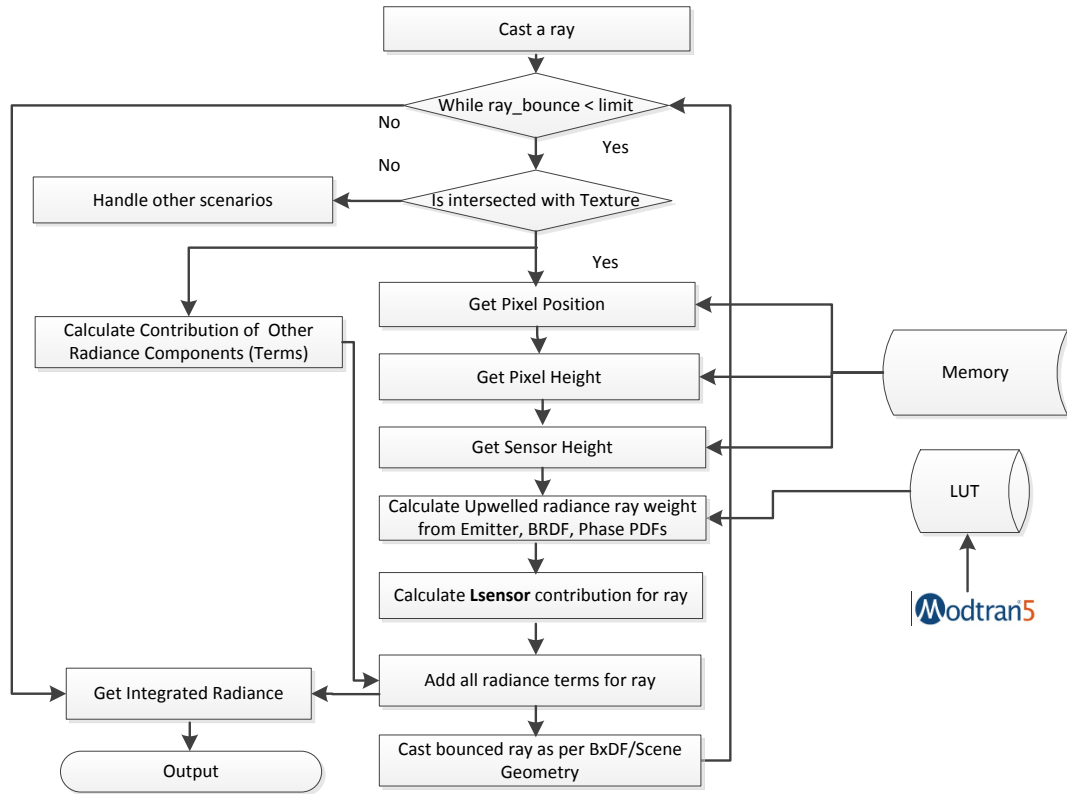


Figure 3.17: Flowchart illustrating recursive ray casting and estimating L_{sensor} during rendering loop.

Radiometric integrator or renderer runs a recursive loop to compute the integrated radiance contribution due to direct and diffuse sources. The light-transport scheme employed in rendering path tracing. A more detailed text about this rendering scheme may be found in Reference [48]. Multiple importance sampling [49] is applied for a direct light source with BRDF/BTDF and diffuse light source with phase function. The Monte-Carlo samples are passed to the camera/sensor module which converts it to rays. Each pixel in the sensor plane is sampled by Poisson/Hammersley sampling and a variable number of rays can be defined in parameters. Increasing the number of rays increases rendering time but provide higher image quality. Distance sampling is performed to estimate the

attenuated transmission due to propagation in the given medium. The solar and diffuse reflected/transmitted radiance contributions are driven by the emitter BxDF and phase sampling and respective PDFs. MIS weights are calculated based on these PDFs in each iteration. The emitter contributions are interpolated from the MODTRAN based results in the database. The phase function employed during rendering is Henyey-Greenstein function.

Rendering is a recursive operation which is repeated for a user-defined ray bounce and terminated at the lowest variance condition guided by means of Russian roulette. The directly reflected radiance does not require more contribution to the recursion than the scattered one. However, the background reflected radiance is also contributed in the recursive loop depths, depending on the background geometry. Figure 3.18 shows the scattering process modelling in a rendering environment.



Figure 3.18: Path tracing based volumetric scattering of monochromatic light rays in presence of aerosols. Scattering event generates a deviation, hence another branch of recursive ray.

A scene is usually split into regions based on dimensions and primitives, where each region is represented by the node in a kd-tree as shown in Figures 3.19 and 3.19. We generated some random points on the basis of which regions are constructed. The traversal of regions is searched through the tree where the location of the region is stored in the tree as a search key. The intersection of rays with the scene is then traced through the search tree.

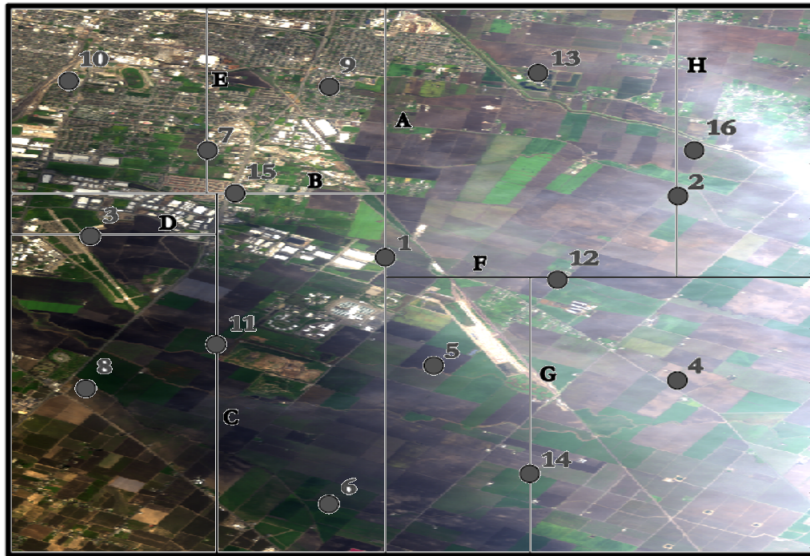


Figure 3.19: Scene geometry is a 3D bounded region, the scene is partitioned into regions by generated random points or primitives. Labelled points are visual depiction of data structure nodes that are handled in kd-trees.

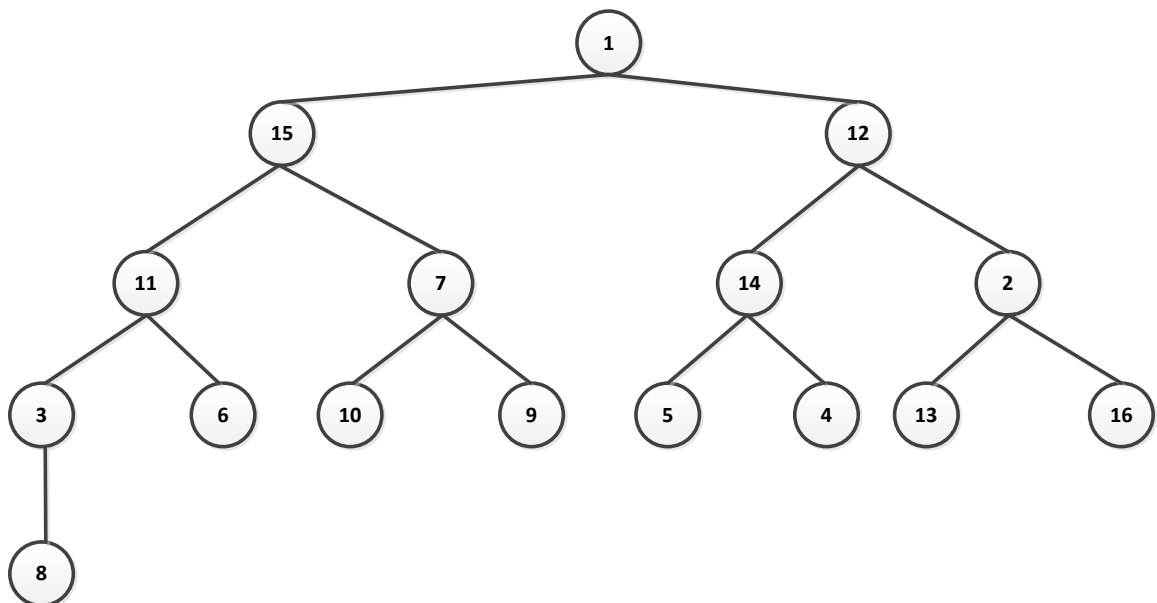


Figure 3.20: kd-tree of partitioned scene area shown in Figure 3.19.

For efficient computing, the scene is also divided into computing blocks where each

block is rendered in parallel. The eventually accumulated radiance for each pixel is computed which is passed to the image reconstruction component. This component is a windowed gaussian reconstruction filter as a default in our simulation, which estimates the pixel intensity based on radiance and applies anti-aliasing filter, together with texture interpolation.

3.5.2 Proposed Adjacency Models for CHIMES Simulator

The adjacency effect reduces with wavelength [61] as it depends on scattering efficiency. In CHIMES we maintain most of DIRSIG radiative transfer equation, however, treatment of I type of rays (adjacency) is handled with two different models. Both of our models adopt DISORT based upwelled scattering, however approach of setting average background albedo $\bar{\rho}_b$ is different in them. Our first and comparably simplistic model is similar to CameoSim.

Based on Equation (2.111), the upwelled scattered radiance into the sensor is therefore expressed as in Equation (3.4). This is the path radiance output from MODTRAN when target albedo is set to zero, during execution.

$$L_{us}^m = \frac{\rho_u E'_s \cos \theta}{\pi}. \quad (3.4)$$

Replacing upwelled scattered term of Equation (2.102) by Equation (3.4) we get Equation (3.5), which is the Lambertian BRDF form of CHIMES Background One-Spectra AEM (BOAEM) model. The final estimation of L_{us}^m during rendering is driven by volumetric scattering. This model is constructed to provide a better comparison of CHIMES with CameoSim, in terms of overall performance, which include several parameters such as sensor model, scene, ray tracing, volumetric scattering and background reflected radiance and so forth.

$$L(\lambda) = \tau_2(\lambda) \left\{ \frac{E'_s(\lambda)}{\pi} \tau_1(\lambda) \rho(\lambda) \cos \sigma + \varepsilon(\lambda) L_T(\lambda) + \rho_d(\lambda) F \left[\frac{E_{ds}(\lambda)}{\pi} + \frac{E_{d\varepsilon}(\lambda)}{\pi} \right] + \rho_d(\lambda) (1 - F) [L_{bs}(\lambda) + L_{b\varepsilon}(\lambda)] \right\} + \frac{\rho_u E'_s \cos \theta}{\pi} + L_{u\varepsilon}(\lambda). \quad (3.5)$$

In the second, more comprehensive, adjacency model, CHIMES adopts the methodol-

ogy similar to that in ATCOR to deduce the upwelling of the adjacency through a sliding window of regions along the horizontal scale of the terrain. Since the extent of the adjacency scattering depends on the density of air within the field of view of the sensor, hence the adjacency is a function of the sensor altitude (h_1) and the elevation of ground (h_2). Through experimental analysis Richter et al. [21] derived an empirical equation to quantify the approximate adjacency range $R(h_1, h_2)$ on the ground as;

$$R = (h_1 - h_2)0.1. \quad (3.6)$$

The above empirical equation is the result of the air density differentiation along elevation z . It has been shown that the density of air $D(z)$ is exponentially reduced above z_d elevation;

$$D(z) = D_o \exp\left(\frac{-z}{z_d}\right), \quad (3.7)$$

where $D(z)$ is air density at altitude z , D_o represents density at sea-level and z_d is the average scale height for the density of air which is 8 Km [21]. The adjacency effect for every pixel in a scene is evaluated by forming a region of interest (ROI) of $(2R \times 2R)$ pixels having the test pixel (target) in the center. According to Equation (3.6) the dimension of the ROI is dependent on the difference of elevation of the test pixel (target) and sensor altitude. Therefore in rugged terrain's ROI dimension is variable across the scene. For each pixel in the scene, CHIMES deduces the path reflectance (ρ_o), adjacent contribution for unit reflectance ($\Delta\rho_u$) and the ROI background reflectance ($\bar{\rho}_{bK}$) for each $(2R \times 2R)$ ROI. We incorporate $\bar{\rho}_{bK}$ into Equation (2.102) by interpolation, therefore ρ_u is calculated for $\bar{\rho}_b = 0.0$ and $\bar{\rho}_b = 1.0$ from Equation (2.111), and it is denoted by ρ_{u0} and ρ_{u1} , respectively which is given in Equation (3.8).

$$\rho_{u0} = \rho_o \quad (3.8a)$$

$$\rho_{u1} = \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dir}}{1 - s} \quad (3.8b)$$

$$\Delta\rho_u = \rho_{u1} - \rho_{u0} = \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dir}}{1 - s}. \quad (3.8c)$$

$\bar{\rho}_{bK}$, inside ROI is calculated by Equation (3.9).

$$\bar{\rho}_{bK} = \sum_{s=1}^M \frac{\rho_s}{R_s^2}, \quad (3.9)$$

where ρ_s is the reflectance of the sample pixel as shown by red pixels depicted in Figure 3.21 (b) and R_s is the distance between target pixel (green/orange) and sample pixels. M is the number of samples in the grid. The upwelled scattered reflectance for the ROI region is therefore computed according to every pixel's position and height by interpolation as given in Equation (3.10), where Subscript K denotes the kernel ROI. The approximate upwelled scattered radiance for the ROI is given in Equation (3.11).

$$\rho_{uK} = \rho_{u0}(1 - \bar{\rho}_{bK}) + \bar{\rho}_{bK}\rho_{u1} \quad (3.10a)$$

$$\rho_{uK} = \bar{\rho}_{bK}\Delta\rho_u + \rho_{u0} \quad (3.10b)$$

$$L_{usK}^m = \frac{(\bar{\rho}_{bK}\Delta\rho_u + \rho_{u0})E'_s \cos\theta}{\pi}. \quad (3.11)$$

We performed simulations by upwelled scattered radiance interpolation between zero and one. In order to have finer interpolation, ρ_u from MODTRAN at more intermediate points between zero and one are taken. Therefore we constructed a set of L_{usK}^m at ten points ranging from 0–1 with an interval of 0.1. This set is denoted as ρ_{uR} , which is given in Equation (3.12), upwelled scattered radiance values of this set are stored in the database.

$$\rho_{uR} = \left\{ \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif}\bar{\rho}_{b0.0}}{1 - s\bar{\rho}_{b0.0}}, \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif}\bar{\rho}_{b0.1}}{1 - s\bar{\rho}_{b0.1}}, \dots, \rho_o + \frac{(\tau_1^{dir} + \tau_1^{dif})\tau_2^{dif}\bar{\rho}_{b1.0}}{1 - s\bar{\rho}_{b1.0}} \right\} \quad (3.12)$$

Suppose ρ_w and ρ_b are given in (3.13).

$$\rho_w = \min\{\rho_{uR} < \max(\rho_{uK})\} \quad (3.13a)$$

$$\rho_b = \max\{\rho_{uR} < \min(\rho_{uK})\} \quad (3.13b)$$

The minimum and maximum value of ρ_{uK} are calculated and closest bounding upper

and lower range value of ρ_{uR} are retrieved for interpolation. This will ensure a better approximation than Equation (3.10). $L_{usK'}^m$ is therefore given in Equation (3.15).

$$\Delta\rho_{u'} = \rho_w - \rho_b \quad (3.14a)$$

$$\rho_{uK'} = \bar{\rho}_{bK}\Delta\rho_{u'} + \rho_{u0} \quad (3.14b)$$

$$L_{usK'}^m = \frac{(\bar{\rho}_{bK}\Delta\rho_{u'} + \rho_{u0})E'_s\cos\theta}{\pi}. \quad (3.15)$$

Once $\bar{\rho}_{bK}$ is computed for the intersecting ray, it is backscattered by coupling the phase function of the atmosphere and the diffuse emitter in ray-tracing by utilizing Monte- Carlo multiple importance sampling [49] and finally values of L_{usK}^m and $L_{usK'}^m$ are estimated. A ray-level computation of upwelled scattered radiance is shown in Figure 3.21 (a). Moreover, the actual model in CHIMES also considers BxDF models in calculating the radiance. It should be noted that, unlike ATCOR, TIAEM does not perform any differencing between target and background reflectance. Contrast reduction is the consequence of volumetric backscattering of the computed upwelled scattered radiance under the kernel, which eventually takes the form of Equation (3.11). The final rendering equation for the TIAEM model (assuming Lambertian BRDF) takes the form of Equation 3.16.

$$\begin{aligned} L(\lambda) = \tau_2(\lambda) \left\{ \frac{E'_s(\lambda)}{\pi} \tau_1(\lambda) \rho(\lambda) \cos\sigma + \varepsilon(\lambda) L_T(\lambda) + \rho_d(\lambda) F \left[\frac{E_{ds}(\lambda)}{\pi} + \frac{E_{d\varepsilon}(\lambda)}{\pi} \right] \right. \\ \left. + \rho_d(\lambda) (1 - F) [L_{bs}(\lambda) + L_{b\varepsilon}(\lambda)] \right\} \\ + \frac{(\bar{\rho}_{bK}\Delta\rho_u + \rho_{u0})E'_s\cos\theta}{\pi} + L_{u\varepsilon}(\lambda). \quad (3.16) \end{aligned}$$

We reused DIRSIG rendering equation to represent the modified upwelled scattered term in our presentation, however, this equation assumes Lambertian BRDF. In our simulator three BRDF models are supported such as Phong, Lambertian and Torrance and Sparrow. In our experiments, we assume Lambertian BRDF for all simulations due to the abundant vegetation in the background of the scene.

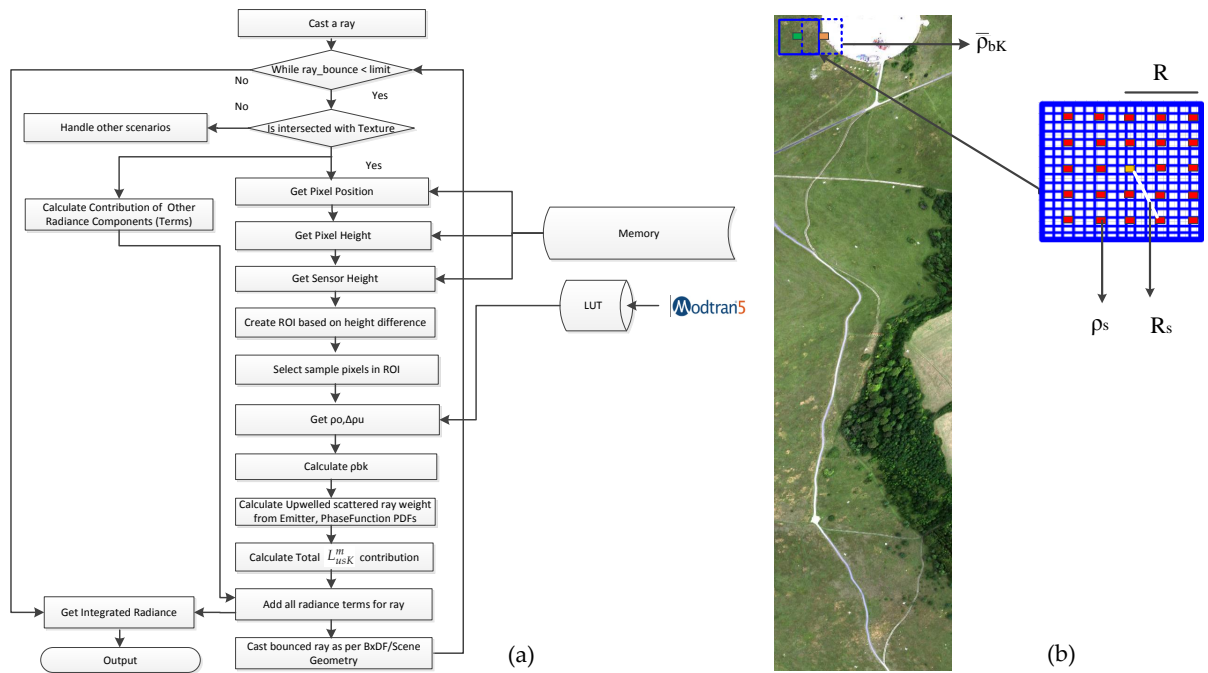
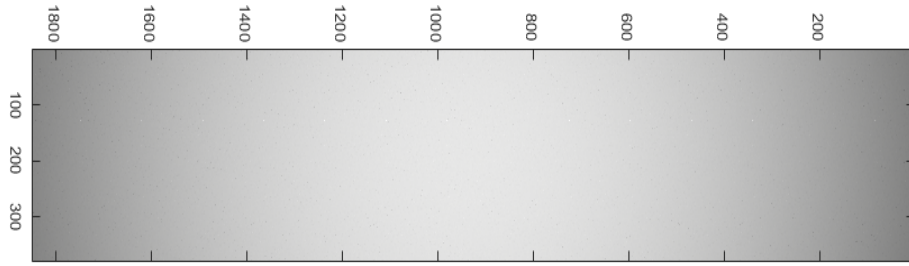


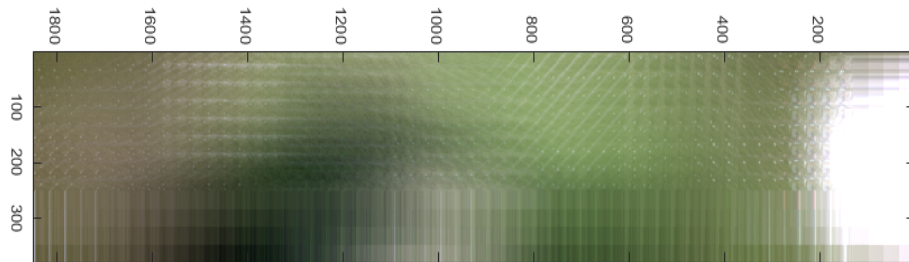
Figure 3.21: (a) The data flow of the proposed TIAEM model which computes the adjacent upwelling radiance by evaluation of volumetric scattering with an spatial ROI. It depends on the difference of altitude between sensor and height of terrain at a given point on texture. (b) shows the evaluation procedure of averaging the background albedo of the ROI by the adjacency pixels (in red) for the test (target) pixel which is in orange/green.

As we learnt that the TIAEM upwelled scattered term is dependent on the sensor altitude, we illustrate the outcome of upwelled scattered radiance for low and high altitude airborne images in shape of Selene and Modesto scenes, respectively.

Selene is an airborne scene with sensor altitude of 0.944 km, due to the less volume of medium between target and sensor the value of R is smaller. Therefore the textural imprint on the upwelled scattered radiance is dominant as shown in Figure 3.22. Similarly, when the sensor is mounted on a sensor with altitude of 20 Km, the value of R is very large and therefore the textural imprint on the upwelled scattered radiance is partially noticeable as shown in Figure 3.23. Both images are generated by CHIMES simulation where all other radiance contributions are discarded except upwelled scattered radiance, during rendering process.

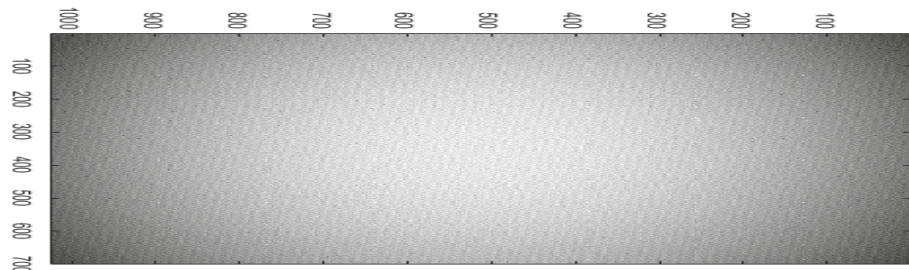


(a) RGB image constructed from **upwelled scattered** term only, for BOAEM adjacency model. It has a grey colour without any texture information.

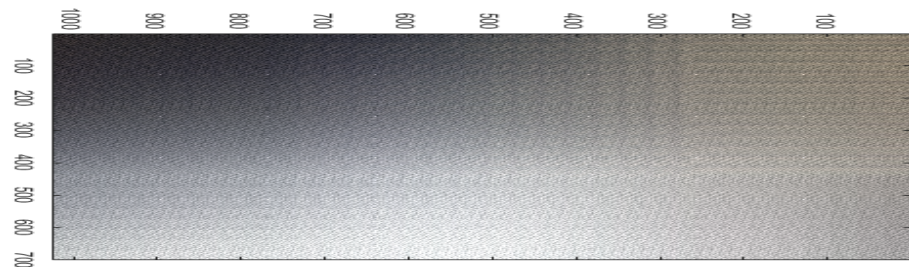


(b) RGB image constructed from **upwelled scattered term** only, for TIAEM adjacency model. A more blurred texture information is visible.

Figure 3.22: Selene Airborne scene; altitude **0.944 Km**. Scene simulated with upwelled scattered term for BOAEM and TIAEM.



(a) Modesto RGB image constructed from **upwelled scattered** term only, for BOAEM adjacency model. It has a grey colour without any texture information.



(b) Modesto RGB image constructed from **upwelled scattered term** only, for TIAEM adjacency model. A more blurred texture information is visible.

Figure 3.23: Modesto Airborne scene; altitude **20 Km**. Scene simulated with upwelled scattered term for BOAEM and TIAEM.

3.6 Automatic Atmospheric Search for Atmospheric Parameters for Scene Simulation

The presence of aerosols or water droplets in the atmosphere is a complicated phenomenon to model as their properties, such as their sizes, densities and their distributions are subjected to many atmospheric factors which makes them highly in-homogeneous along the solar zenith and azimuth dimensions. Given the upwelling radiance of a scene, there is no method to know exactly what are the aerosols and their density distributions over the measured site because it is an inverse problem and the solution is likely to be non-unique. To simulate a scene, one will need the knowledge of the material characteristic on the ground, and more importantly the atmosphere property over the scene. The latter input is normally ‘guessed’ from the data provided by the weather stations perhaps with the help from radiosonde and so forth. Among the list of atmosphere parameters which is outlined in Table 3.1, some relatively more important ones such as the optical thickness and water droplets dimensions in the cloud can be modelled within certain limits given by the seasonal model of the test site [18]. Studies [68] have shown that light’s extinction by the presence of clouds is dominantly affected by the effective radius of droplets (r) and their densities (C_w) in the cloud.

3.6.1 MODTRAN Parameters Types

MODTRAN provides a list of parameters to generate suitable atmospheric data based on season, atmospheric medium constituents, clouds, sensor orientation and altitude, observation parameters, volcanic background type and light source types. We classify parameters in two types one is user-defined and the other is searched. User-defined parameters are those that match the meta-information about a scene that is to be simulated, such as date, time, location, sensor orientation, season, sensor altitude, average scene temperature at the time of the acquisition, background adjacency albedo. The other more complicated parameters that are typically hard to find are aerosol’s thickness, water vapour concentration and aerosol types. The aerosol type is also typically found through a pre-defined rule-base depending on the sensor’s altitude.

Table 3.1: MODTRAN parameters and their respective value for the Selene scene. Values in boldface are the variables that are searched for different values to fit the closest radiance. Other values are fixed for a specific scene according to the acquired scene's meta information and date/time.

MODTRAN Parameter	Value [Searched Value]
Seasonal model	Sub-arctic summer
Water vapour column (gm/m ³)	Variable (List) [0.223]
CO ₂ (ppm)	400
Boundary Temperature (K)	283
Background Adjacency Albedo	Grass scrub (Spectral)
Aerosol Thickness	Variable (List) [0.024 Km]
Spectral Range	0.414 μm –2.51 μm
Scattering	DISORT Scaled, 8 streams
Sensor Orientation	nadir
Observation Parameters	12/08/2014, 13:04 (GMT), Solar Zenith : 37.5° Azimuth : 199.88°, Sensor Altitude : 0.944 Km, Target Altitude : 0.001 Km
Cloud base	0.610 Km
Cloud Types	Stratus, (Cumulus, Stratocumulus, Altostratus, Cirrus, Rain, User-defined)
Vertical Structure Algorithm	None
Volcano Background	Normal Volcano Background
Light Source	Sun, (Moon)

The optical transmittance τ of a slab of cloud with thickness R , effective radius of droplets in the cloud r , density of droplets in the cloud C_w and density of water ρ ($\rho = 1.0 \text{ gm/cm}^3$) has been found analytically in the form of Equation (3.18) [57, 68].

$$\tau = \exp(-\delta) \quad (3.17)$$

$$\delta = \frac{3C_w R}{2\rho r}. \quad (3.18)$$

Thus, the above equation gives a first-order estimation of the optical transmittance $\tau(C_w, R, r)$ of the cloud. For a scene with known materials on the ground and when it subjects to solar irradiation, the albedo of the ground or the radiance of some specific materials in the scene can be evaluated analytically through Equation (3.18) for a given set of

droplet parameters (C_w, R, r), For low altitude clouds having Stratus profile, MODTRAN assumes ($r = 8.33 \mu\text{m}$). A set of the albedo can then be evaluated under systematic variation of these droplet parameters as according to Equation (3.18). After this set of albedo is convolved with the sensor characteristics, they can then be monitored through a distance metric measurement compared to the ground truth radiances of the scene. In a nut shell the atmospheric search algorithm is a subroutine to obtain the ground or material albedo under the different configuration of cloud (C_w, R, r) and a subsequent call of MODTRAN for the radiative transfer evaluation. A flowchart of the atmospheric search is given in Figure 3.24. The experimental setup during GT data acquisition and the results of the search algorithm are presented in Section 6.2.1.

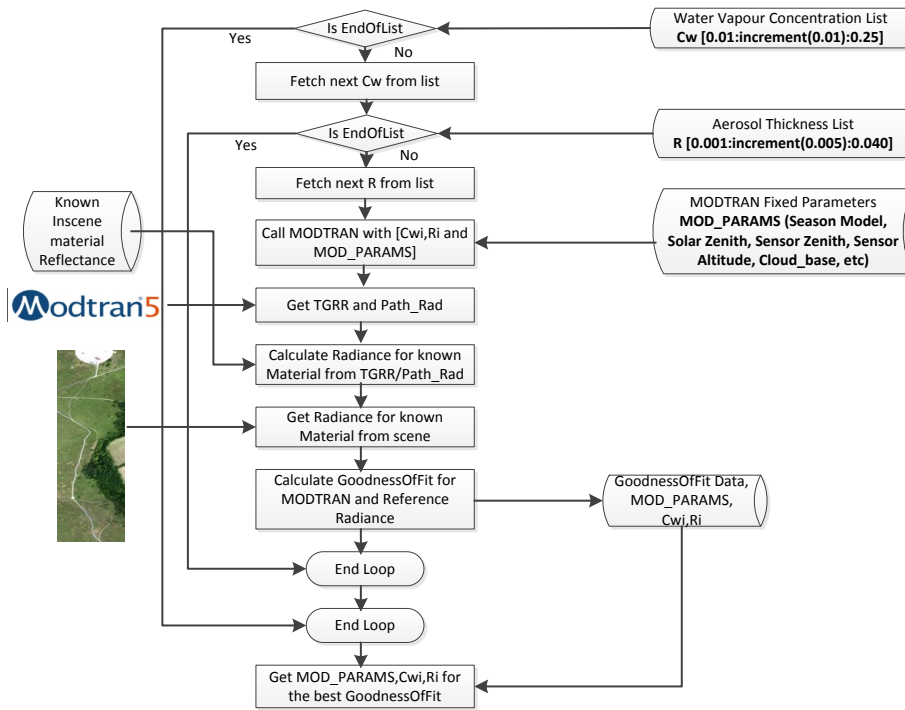


Figure 3.24: Flowchart of searching underlying atmospheric conditions parameters.

In this section, we discussed adjacency models included in CHIMES. Concisely, the BOAEM is a simplistic adjacency model which evaluates the upwelling by using one spectral reflectance. Other simulators such as the CameoSim has also employed a similar model for adjacency modelling. However, the TIAEM may represent a more comprehensive adjacency model in which it evaluates the local adjacency in upwelling radiance within several small ROIs across the entire scene.

3.6.2 Automatic Search for Atmospheric Parameters

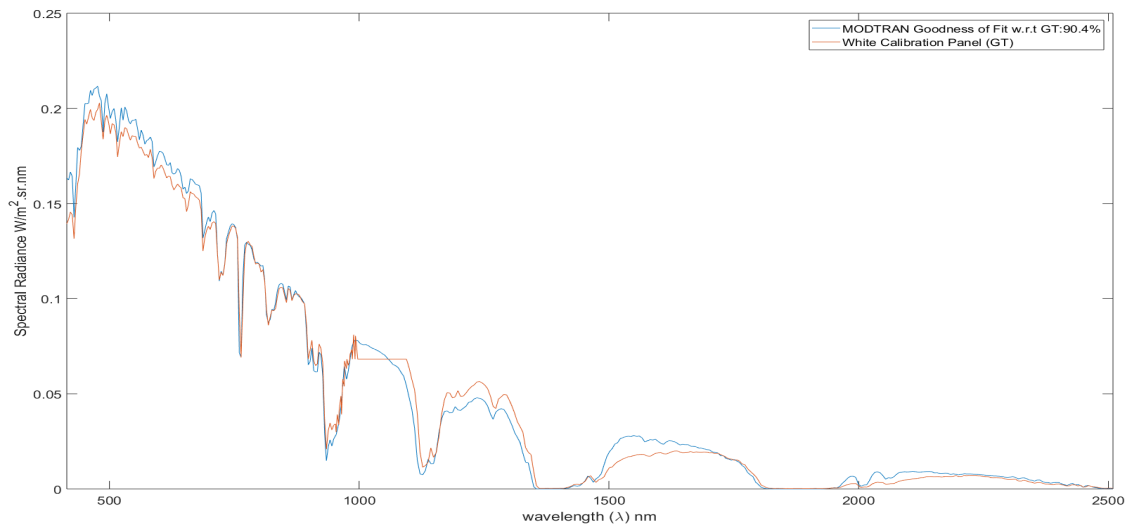
During the search process, MODTRAN is input with several parameters, few of them are shown in Table 3.1. In our experimental data, sensor altitude is higher than cloud base and presence of thin cloud below the sensor is also reported [10], therefore we will vary the aerosol optical thickness below the sensor to search the approximate underlying atmospheric condition in the scene.

In each iteration MODTRAN's L_w that is, accumulated total ground reflected radiance and path radiance is calculated and goodness of fit is recorded. Estimate of searched white panel radiance L_w and black panel radiance L_b is shown in Figure 3.25, that possesses the highest goodness of fit, that is 90.4% of match in terms of NRMSE for L_w and 55.26% for L_b , respectively. These results manifest good atmospheric search in terms of total ground reflected radiance, however, estimation of path radiance is relatively worse. If path radiance estimated in the search process is lower than the GT panel then it may result in higher contrast than the GT. As the search method only employs MODTRAN at-sensor radiance estimate, the effect of volumetric scattering is not accounted for. At rendering time, volumetric scattering may either reduce the error in the estimate of L_b or increase it depending on the choice of phase function and backscattering parameter, such as the variable g in Henyey-Greenstein function [48]. However, due to good estimation of the TGRR, we have selected searched parameters ($C_w = 0.22 \text{ gm/m}^3$ and $R = 0.024 \text{ Km}$) and fixed parameters given in Table 3.1 for our overcast condition simulations. In the case of atmospheric search by using white and black panel, it is easier to segregate the path radiance and TGRR, therefore these targets were employed. A similar search based on orange perspex yielded ($C_w = 0.22 \text{ gm/m}^3$ and $R = 0.023 \text{ Km}$), which is very close to white and black panel's parameters.

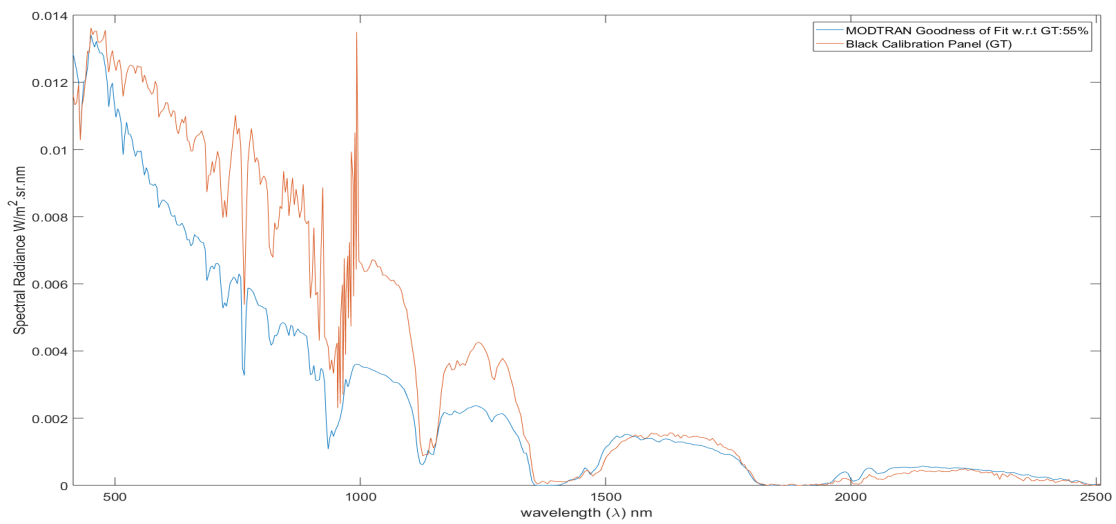
3.6.2.1 Drawbacks of Atmospheric Parameters Searching

The atmospheric parameters of MODTRAN listed in Table 3.1 are searched to best fit the radiance of known reflectance material by iterating MODTRAN runs. Unfortunately, MODTRAN does not provide the facility to incorporate DEM data for a terrain. Therefore the parameter a searched for an absolute flat terrain, therefore the searched parameters based atmospheric data is expected to yield more error in CHIMES environment when rendering is performed on a DEM terrain. It would accumulate higher intensity at each pixel due to the capture of ray bouncing in a path-tracing environment. To keep the search

computationally efficient higher priority is given to *Direct* and *Diffuse* reflected radiance which matching the reduced NRMSE error than the *Path* radiance. This is apparent in the results of Figure 3.25, where L_w has a higher match (90%) with the in-scene radiance and L_b has lower match (55%), only. The rationale behind taking only L_w into account in our empirical experience with the output of MODTRAN's path radiance which contributes between 10%–15% in at-sensor radiance output. Details about material's used for calibration panel and their respective reflectivity is given in Tables 5.2 and 5.3.



(a) White Panel Estimation



(b) Black Panel Estimation

Figure 3.25: Atmospheric search based White and black panel estimates.

Chapter 4

Shadow Detection

4.1 Shadow Detection in an Hyperspectral image

High-Fidelity hyperspectral imagery contains crucial spatial and spectral information of a given scene. Presence of shadows causes significant challenges for both satellite and airborne data analyses. Shadows cast by scene geometry or clouds cause hurdles in remote-sensing data analyses, including inaccurate atmospheric compensation, biased estimation of Normalized Difference Vegetation Index (NDVI), confusion in land cover classification, and anomalous detection of landcover variation. Therefore, shadows are a significant source of noise in Hyperspectral Image (HSI) data, and their detection is a vital pre-processing step in most analyses [69, 70].

Over the years, various methods of shadow detection proposed are object-based shadow detection methods which classify clouds, their shadows, and non-shadowed regions by applying image segmentation at different bandwidth images of HSI imagery, e.g., [71, 72], and color invariance-based shadow detection methods create RGB in invariant colour space and exploit it for classification as in [73]. Some algorithms use band indices to detect shadows in an HSI image [74, 75]. Another class of algorithms require an a priori Digital Surface Model (DSM) [76] or Terrestrial Laser Scanning data together with the HSI image to find shadows cast from scene geometry [77].

[78] categorised shadow detection algorithm into four classes; a) Chromacity-based methods b) Physical methods c) Geometry-based methods and d) Texture-based methods. They listed almost thirty algorithms in all these categories. None of these categories is model-based and following Radiative Transfer methods. Therefore, the proposed algorithm brings RT-based model-based approach which is not attempted to the extent in the existing literature. Most shadow detection algorithms are following chromaticity and texture-based approaches. We are particularly interested in algorithms that are processed

on the multi-spectral image. There are several research proposals with this attribute such as Beril et al. [79] proposed a colour-invariant function for detecting buildings. Once buildings are detected, then they used the grayscale histogram of the image to detect shadows around the building using the Otsu algorithm [80], their algorithm is referred to as Beril's algorithm in the results. In another contribution, Teke et al. [81] proposed a false colour space consisting of red, green and near-infrared (NIR) bands. They dropped the blue colour because it contains scattered light and removing it will increase the contrast between shadowed and non-shadowed regions and will facilitate detection. They have named their algorithm the Land Use Land Cover classification method, or LULC in their code. Therefore, we will refer to their work as LULC algorithm in our analysis. Sevim et al. [82] modified the C_1, C_2, C_3 color space [83] to accommodate the NIR band, and supplemented it to become the C'_1, C'_2, C'_3, C'_4 color space. We refer to their work as RGBN algorithm. Gevers et al. [84] proposed colour-invariance functions to separate shadow and non-shadow regions; their algorithm is referred to as Gevers' algorithm [85].

The approaches mentioned above are limited to use in particular bands and does not use the complete hyperspectral data. Our algorithm may ideally use HSI data and can be down-scaled to multispectral imagery only in cases where data-acquisition sensor response is known. Spectral response is typically available for most Earth-observation satellites. Our algorithm does not address RGB images because QUAC[63] may not be applied to retrieve reflectance from RGB images. More importantly, our algorithm provides a mathematical foundation for shadow detection based on the RT model and highlights the sources of errors.

The application of Empirical Line Method (ELM) for hyperspectral Atmospheric Compensation (AC) premises the underlying linear relationship between a material's reflectance and appearance. ELM solves the Radiative Transfer (RT) equation under specialized constraint by in-scene white and black calibration panels. The reflectance of material is invariant to illumination. Exploiting this property, we articulated a mathematical formulation based on the RT model to create cost functions relating variably illuminated regions within a scene. In this research, we propose multi-layered regression learning-based recovery of radiance components, i.e., total ground-reflected radiance and path radiance from reflectance and radiance images of the scene. These decomposed components represent terms in the RT equation and enable us to relate variable illumination. Therefore, we assume that the Hyperspectral Image (HSI) radiance of the scene is provided and AC can be processed on it, preferably with QUICK Atmospheric Correction

(QUAC) algorithm. QUAC is preferred because it does not account for surface models. The output from the proposed algorithm is an intermediate map of the scene on which our mathematically derived binary and multi-label threshold is applied to classify shadowed and non-shadowed regions. Results from a high and low altitude airborne NADIR imagery are shown in this report. Ground truth (GT) is generated by ray-tracing on a LIDAR-based surface model in the form of contour data, of the scene. Comparison of our results with GT implies that our algorithm's binary classification shadow maps outperform other existing shadow detection algorithms in true-positive, which is the detection of shadows when it is in the ground truth. It also has the lowest false-positive i.e., detecting the non-shadowed region as shadowed, compared to existing algorithms. Contents of this chapter are based on Reference [18].

4.2 Radiative Transfer Model-Based Relationship between Shadowed and Non-Shadowed Regions

The scope of this research is within optical shadowing, and therefore, the subsequent discussion does not consider thermal radiance and shadowing and their relative terms in Radiative Transfer (RT) equations. This section is divided into two parts—the first presents a general description of the RT equation highlighting relevant parameters and elaborating the sources of errors and their impact on this work, and the second establishes the proposed general relationship between variably illuminated regions based on the RT equation [18].

4.2.1 Proposed Radiative Transfer Model-Based General Relationship between Variably Illuminated Regions

ELM represents the at-sensor radiance from in-scene white and black calibration panels. Radiance reflected from the white panel L_w and black panel L_b is deduced from Equation (2.112) and shown in Equations (4.1) and (4.2), respectively.

$$L_w = L_p + \frac{\tau E_g(0)/\pi}{1 - s\rho} \quad (4.1)$$

$$L_b = L_p \quad (4.2)$$

In Equation (4.1), L_w radiance includes both path radiance and total ground-reflected radiance, which are the first and second terms, respectively. We introduce two parameters α and β in Equation (4.3) that represent the total ground-reflected radiance and path radiance, respectively, and brings the convention to Equation (4.3), which is called the ELM equation.

$$\rho = \alpha L_s + \beta \quad (4.3)$$

where α and β are,

$$\alpha = 1/(L_w - L_b) \quad (4.4)$$

$$\beta = L_b/(L_w - L_b) \quad (4.5)$$

Reflectance ρ is independent of illumination conditions, and enables us to rephrase Equation (4.3) as Equation (4.6), emphasizing only parameters α and β .

$$f(\alpha, \beta) = L_s = \frac{\rho - \beta}{\alpha} \quad (4.6)$$

We rewrite Equation (4.3) for shadowed (sub-scripted S) and non-shadowed (sub-scripted NS) regions of the scene as Equations (4.7) and (4.8), respectively.

$$\rho = \alpha_{NS} L_{S_{NS}}(\rho) + \beta_{NS} \quad (4.7)$$

$$\rho = \alpha_S L_S(\rho) + \beta_S \quad (4.8)$$

Equating Equations (4.7) and (4.8) we get Equation (4.9).

$$L_{S_{NS}} = \frac{\alpha_S L_S(\rho) + \beta_S - \beta_{NS}}{\alpha_{NS}} \quad (4.9a)$$

$$L_{S_{NS}} = \frac{\alpha_S L_S(\rho) + \Delta\beta}{\alpha_{NS}} \quad (4.9b)$$

where $\Delta\beta = \beta_S - \beta_{NS}$. Equation (4.9) is rephrased as Equation (4.10).

$$L_{S_{NS}} = \gamma L_S(\rho) + \delta \quad (4.10)$$

where

$$\gamma = \alpha_S / \alpha_{NS} \quad (4.11a)$$

$$\delta = \Delta\beta / \alpha_{NS} \quad (4.11b)$$

Equation (4.10) shows γ and δ that are two unknown parameters responsible for illumination variability between shaded and non-shaded regions. Ideally, if $\gamma=1$ and $\delta=0$, there is no variability in illumination across the scene.

4.3 Proposed Multi-Layered Regression Learning Algorithm

In the previous section, a general relationship between illumination under shadowed and non-shadowed region within an HSI image is established. Estimation of discriminant parameters α_{NS} , β_{NS} , α_S , and β_S is vital for good detection. We divide our learning algorithm into three phases: (i) regression learning; (ii) feature learning; and (iii) classification, as shown in Figure 4.1.

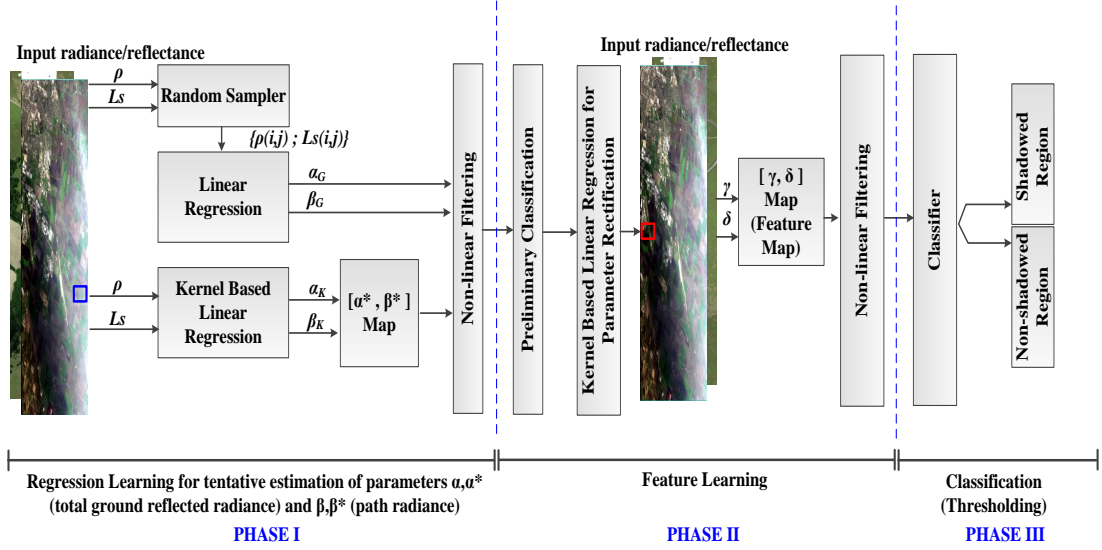


Figure 4.1: Components of multi-layered regression learning algorithm has three phases: (i) regression learning; (ii) feature learning; and (iii) classification. Inputs are radiance and reflectance cubes that are randomly sampled to find neutral parameter estimates for α_G and β_G . A kernel of size 3 used for kernel-based linear regression parameters are α_K and β_K , representing a more localized and homogeneous (either shadowed or non-shadowed regions) estimate. In Phase II, more discriminating parameters are found in the second layer of regression learning, which rectifies parameters estimated by the previous phase. Two non-linear filter layers are shown. Eventually, the classifier layer segregates shadowed and non-shadowed regions.

During learning, Equation (4.3) becomes Equation (4.12), where $\tilde{\rho}$ is the approximate reflectance at any given search iteration, while reflectance computed from QUAC is the reference reflectance, referred in Equation (4.13). Therefore, the cost function $J(\alpha, \beta)$, to minimize is given in Equation (4.14), which is illustrated in Figure 4.2.

$$\tilde{\rho} = \alpha Ls + \beta \quad (4.12)$$

$$\rho_{QUAC} = \rho \quad (4.13)$$

$$J(\alpha, \beta) = \min_{\alpha, \beta} \{\rho - \tilde{\rho}\} \quad (4.14)$$

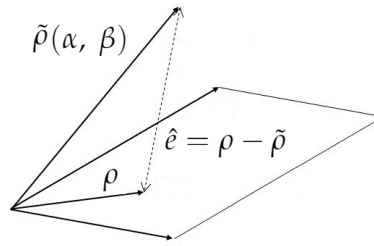


Figure 4.2: Regression learning reduces error \hat{e} to find parameters α (total ground-reflected radiance) and β (path radiance).

A complete flowchart of multi-layered learning is shown in Figure 4.6.

4.3.1 Regression Learning Phase

This phase is denoted as Phase I in Figure 4.1 and is further sub-divided into two steps:

- *Global Search*: Search for α_G and β_G estimates of Equation (4.15) on random samples drawn from the whole image giving a global search across the scene, as shown in Figure 4.1.
- *Local Search*: Create a 3×3 kernel and search for parameters α_K and β_K in the kernel only i.e., localized search.

4.3.1.1 Global Search

Satellite/airborne images cover a larger landscape where the number of bands is more contiguous in HSI images. Higher-resolution images have immediate implications of an increase in both computing and memory requirements. To reduce these requirements, we introduce random sampling on the whole image. A random sampler selects several samples from the whole image and regression learning is performed on these samples to estimate the Empirical Line Method (ELM) parameters. As the input image and the samples (pixels) contain both shadowed and non-shadowed region's pixels therefore parameters α_G and β_G are estimated also represent both shadow and non-shadow pixels TGRR estimates. As we rationally assume that an image has more non-shadow than the shadow

pixel, therefore α_G and β_G are biased towards non-shadow regions and their TGRR. Equation (4.3) for global search case is reformulated in Equation (4.15).

$$\rho_G = \alpha_G L_S G + \beta_G \quad (4.15)$$

An estimate of L_w and L_b found in this phase is shown in Figure 4.3. It is shown that the recovery of L_w employing regression deviates by 20% NRMSE. The panels are placed on a large concrete hard target therefore it does not show large adjacency effect at 1100 nm where vegetation has peak reflectance. As regression samples are selected randomly and the scene is vegetation abundant therefore large error is observed at 1100 nm. The recovered radiance for L_b shows very large deviation i.e. up to 76%. The radiance looks similar to vegetation's radiance, a careful analysis concluded that even the radiance magnitude of black calibration panel was higher than some vegetation under the shadow. This led to the wrong estimation of L_b during regression phase.

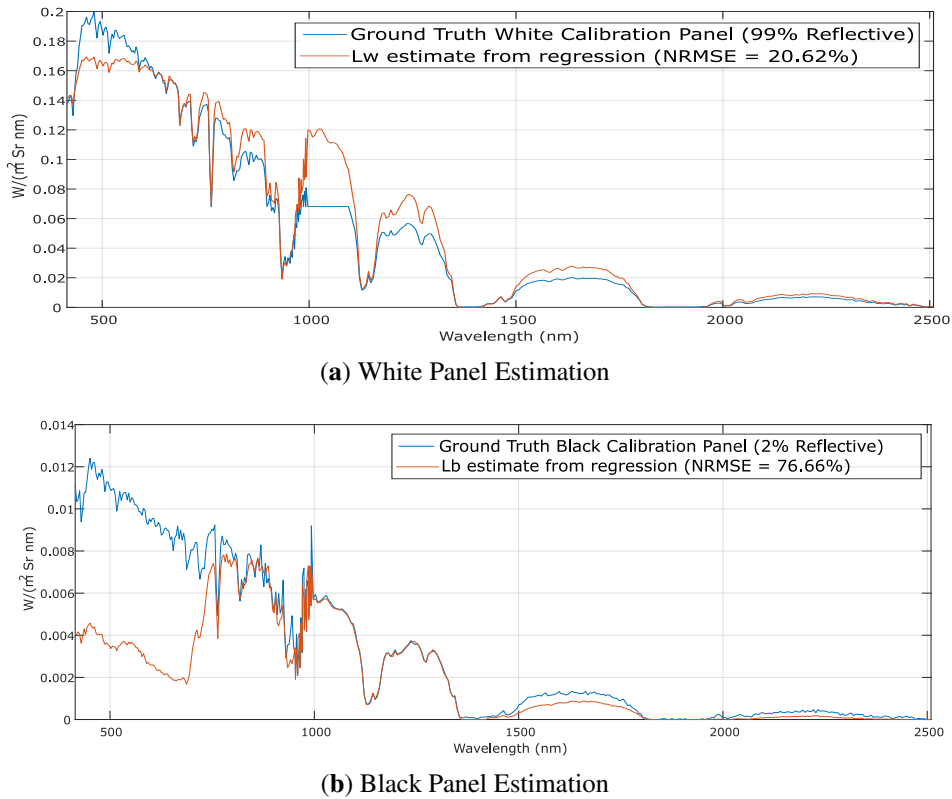


Figure 4.3: White and black panel estimates from global search.

4.3.1.2 Local Search

In this part of regression learning, a 3×3 kernel is used. A smaller kernel provides a rationale for the assumption that pixels under the kernel are homogeneously illuminated. The correctness of this assumption is further reinforced for higher-resolution images that possess lower ground-sampling distance (GSD). Due to learning on a sliding kernel, this search is more time-consuming than the global one. Outputs from this step are parameter (α_K, β_K) maps. An estimate of Lw_K and Lb_K for 465.611 nm is shown in Figure 4.4. In this case, we reformulate Equation (4.3) as Equation (4.16).

$$\rho_K = \alpha_K Ls_K + \beta_K \quad (4.16)$$

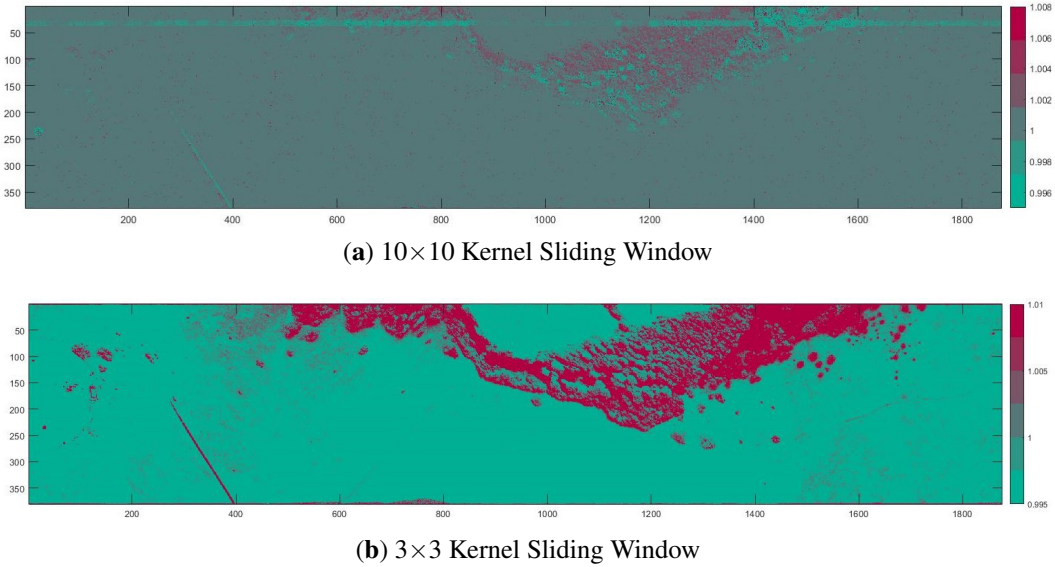


Figure 4.4: Intermediate Map extracted by proposed method on coarse (10×10) and fine (3×3) sliding window kernel. To highlight the shadow region $1/\tilde{t}$ is computed from Equation 4.17, whose values are shown in the bar. The magnitude above 1.0 is classified as shadowed region. As \tilde{t} is a ratio of same quantity therefore it has no units.

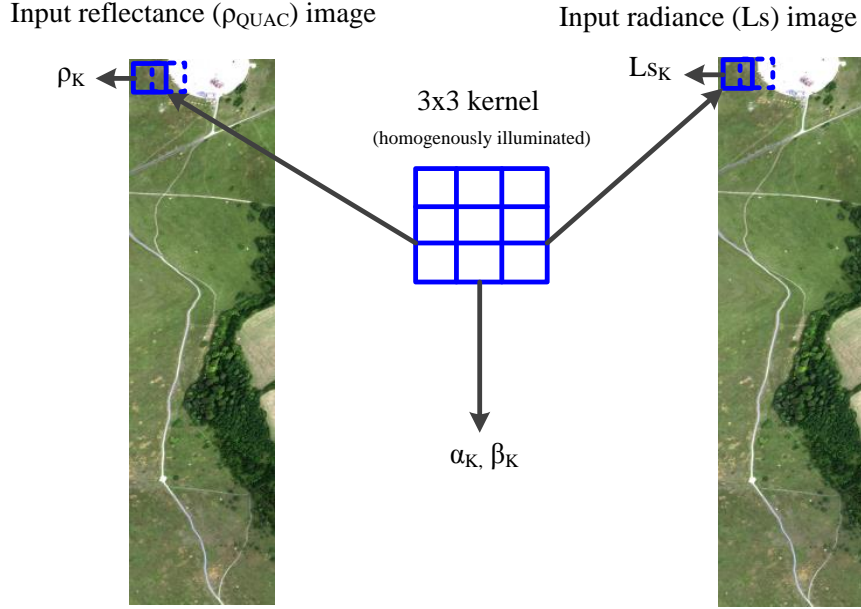


Figure 4.5: Kernel-based regression performs search for localized parameters α_K and β_K by pixels that are only within kernel.

4.3.2 Feature-Learning Phase

In the previous phase, we found both global and local parameters, although we assume that since a smaller kernel has homogeneous illumination it is yet to be ascertained whether it is shadowed or non-shadowed. This phase will establish the discriminant function, first for tentative and then for the final classification. Components of feature-learning phase in Figure 4.1 are described in the subsequent sections.

In this phase, a ratio between $f(\alpha_K, \beta_K)$ and $f(\alpha_G, \beta_G)$ is calculated by Equation (4.17), which yields approximate threshold \tilde{t} .

$$g(\alpha_G, \beta_G, \alpha_K, \beta_K) = \frac{f(\alpha_K, \beta_K)}{f(\alpha_G, \beta_G)} = \tilde{t} \quad (4.17)$$

$$\tilde{t} = \begin{cases} > 1 & , \quad f(\alpha_K, \beta_K) = f(\alpha_{NS}, \beta_{NS}), \quad (\text{Non-shadowed}) \\ < 1 & , \quad f(\alpha_K, \beta_K) = f(\alpha_S, \beta_S), \quad (\text{Shadowed}) \end{cases} \quad (4.18)$$

The kernel region is assigned a shadow or non-shadow label based on Equation (4.18). If the value of t is greater than one, then the region under the kernel is more likely to be non-shadowed than otherwise. The above-mentioned threshold \tilde{t} holds under the intuitive assumption that the scene has more non-shadowed regions than shadowed ones. This provides a preliminary classification as shown in Figures 4.1 and 4.6, and leads us to either $(L_{NS}, \alpha_{NS}, \beta_{NS})$ or (L_S, α_S, β_S) of Equation (4.9a), as per our assumption. The selection of either of these parameter sets is shown as two potential flows in Figure 4.6.

In practice, \tilde{t} is noisy data across the optical spectrum. Therefore, non-linear filtering is applied to it for smoothing and creating an intermediate map. As filtering is applied to both global and kernel thresholds \tilde{t} and t_K , respectively, it is further discussed in Section 4.3.3.

4.3.2.1 Preliminary Classification

Global radiance L_S is estimated by inclusion of both shadowed and non-shadowed regions. L_K is however assumed to be either of them. A global and local version of Equation (4.6) is given as

$$f(\alpha_K, \beta_K) = L_K = \frac{\rho_K - \beta_K}{\alpha_K} \quad (4.19a)$$

$$f(\alpha_G, \beta_G) = L_G = \frac{\rho_G - \beta_G}{\alpha_G} \quad (4.19b)$$

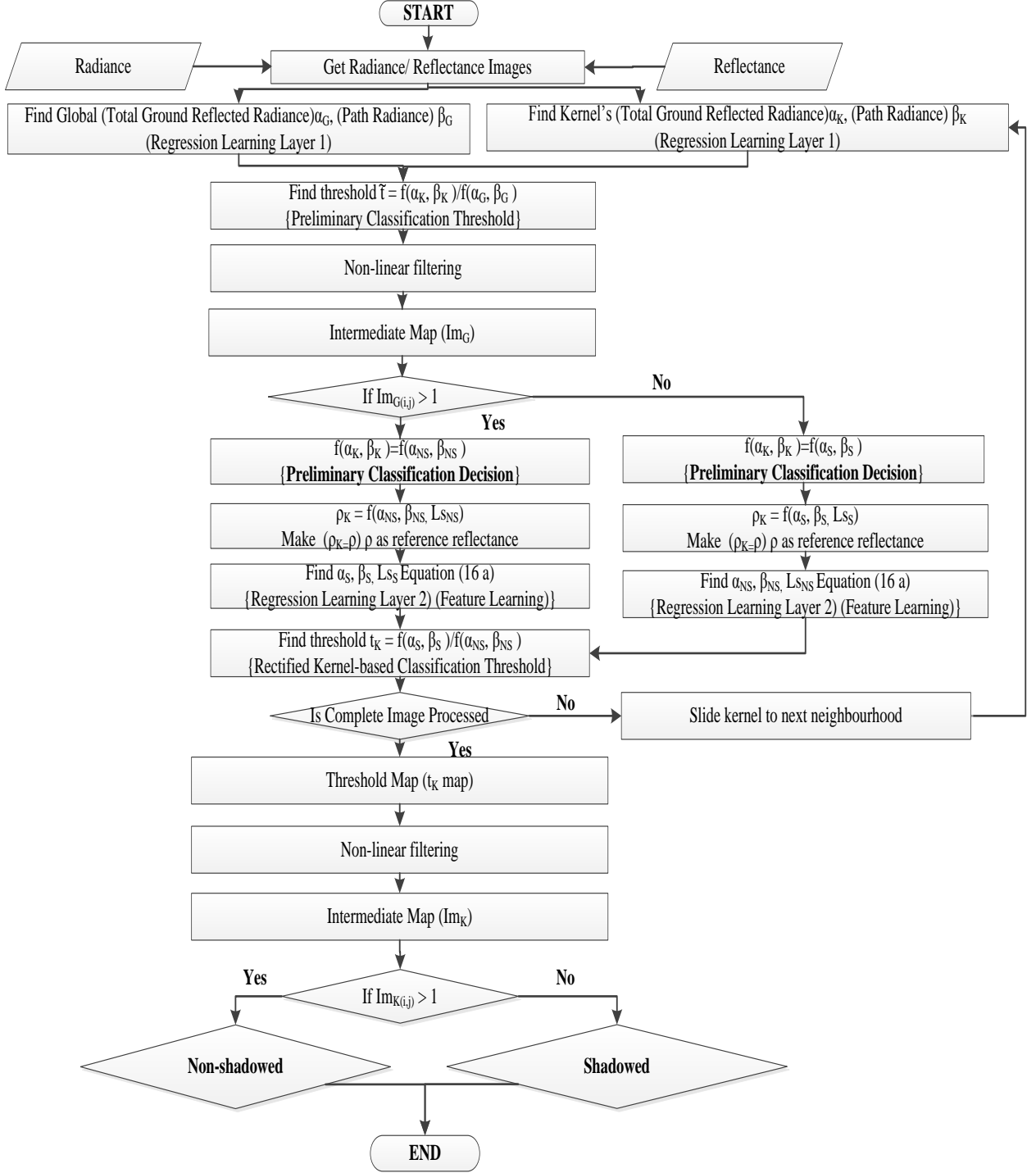


Figure 4.6: Flowchart of the proposed multi-layered regression learning algorithm. Parameters α, β with subscript “ G ” denote global, and “ K ” denotes local (under the kernel). Moreover, the parameter subscript “ NS ” denotes non-shadowed and “ S ” denotes shadowed regions. ρ stands for reflectance, and t denotes threshold. In global search several pixels are randomly selected for regression learning and retrieval of TGRR. In local search a kernel is slid over the image where pixel observations are selected from the active kernel position.

4.3.2.2 Regression Learning for Parameter Rectification

To create a more reliable discriminant threshold function than that of Equation (4.18), we need to establish a relationship between our processed parameters to bring it in the shape of Equation (4.9). Global and local processing has provided us with a mathematical basis for tentative classification, which is plausible under the laws of physics. We suppose that preliminary classification labels the kernel region as non-shadowed which implies that we found $(L_{S_{NS}}, \alpha_{NS}, \beta_{NS})$; however $(L_{S_S}, \alpha_S, \beta_S)$ of Equation (4.9)a are yet to be determined for the region under the kernel. We rewrite Equation (4.9)a, for convenience.

$$L_{S_{NS}} = \frac{\alpha_S L_{S_S}(\rho) + \beta_S - \beta_{NS}}{\alpha_{NS}} \quad (4.20)$$

$L_{S_S}, \alpha_S, \beta_S$ are determined by another layer of regression learning. The cost function $Q(L_{S_S}, \alpha_S, \beta_S)$, to minimize is given in Equation (4.14), which is illustrated in Figure 4.7.

$$\tilde{\rho} = \alpha L_{S_S} + \beta_S \quad (4.21)$$

$$\rho = \alpha_{NS} L_{S_{NS}} + \beta_{NS} \quad (4.22)$$

$$Q(L_{S_S}, \alpha_S, \beta_S) = \min_{L_{S_S}, \alpha_S, \beta_S} \{\rho - \tilde{\rho}\} \quad (4.23)$$

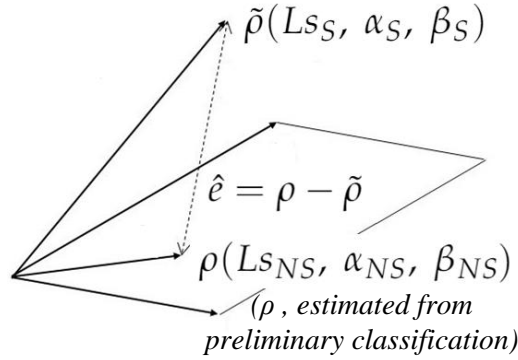


Figure 4.7: Regression learning reduces error \hat{e} to find parameters α_S (total ground-reflected radiance in the shadow region) and β_S (path radiance in shadow region) and L_S (total radiance under shadow region).

This learning is performed by Algorithm 2. After this learning phase, we estimated

all unknown parameters of Equation (4.20), therefore we may rewrite Equation (4.17) as Equations (4.24) and (4.25). Please note that this equation is only for the kernel region, which has the same reflectance ρ and has shadow and non-shadow parameters instead of global and local.

Algorithm 2 Gradient-descent algorithm for global/kernel-based search.

- 1: Let HSI Radiance Image be “L” with “s” samples and “b” bands, and its reflectance estimated by Run QUAC to find reflectance R
 - 2: Assign outputs L_w and L_b as two zeros vector of “b”
 - 3: Let stepSize be 0.01 with a decay of 0.995
 - 4: **while** ($\Delta L_w \leq 1 \times 10^{-10}$ and $\Delta L_b \leq 1 \times 10^{-10}$) **do**
 - 5: Select 2 pixels at random
 - 6: Estimate δL_w and δL_b for selected pixels by ELM equation
 - 7: $L_w = (1 - \text{stepSize}) \times L_w + \text{stepSize} \times \delta L_w$
 - 8: $L_b = (1 - \text{stepSize}) \times L_b + \text{stepSize} \times \delta L_b$
 - 9: stepSize = stepSize \times decay
-

It is extremely important to note that if preliminary classification finds $(\alpha_K, \beta_K) = (\alpha_{NS}, \beta_{NS})$ then t takes the form of Equation (4.24). If it finds $(\alpha_K, \beta_K) = (\alpha_S, \beta_S)$ then t takes the form of Equation (4.25).

$$g(\alpha_{NS}, \beta_{NS}, \alpha_S, \beta_S) = \frac{f(\alpha_{NS}, \beta_{NS})}{f(\alpha_S, \beta_S)} = t \quad (4.24)$$

$$g(\alpha_S, \beta_S, \alpha_{NS}, \beta_{NS}) = \frac{f(\alpha_S, \beta_S)}{f(\alpha_{NS}, \beta_{NS})} = t \quad (4.25)$$

Replacing both numerator and denominator of Equation (4.24) and 4.25 by right-hand side of Equation (4.6) in their respective form, we get Equation (4.26) and Equation (4.27).

$$t = \frac{\alpha_{NS}(\rho - \beta_S)}{\alpha_S(\rho - \beta_{NS})} \quad (4.26)$$

$$t = \frac{\alpha_S(\rho - \beta_{NS})}{\alpha_{NS}(\rho - \beta_S)} \quad (4.27)$$

Threshold t is a better estimate than \tilde{t} , because both parameters are computed for the region under the kernel. This process is termed as rectification in Figures 4.1 and 4.6. When the kernel slides through the image, it estimates t at each iteration. On completion, it creates an intermediate map for the whole image. As we discussed in Section 4.3.2.1, both \tilde{t} and t are computed for all bands of HSI image and tend to get very noisy in some spectral bands. This problem is tackled by filtering, which is covered in Section 4.3.3.

4.3.3 Filtering

Filtering is a supplementary but vital step for the performance of detection. This is the third and final layer of regression learning. Here we estimate an activation function for a non-linear filter based on Equation (4.28a). The cost function $J(x, t)$ finds the value of threshold scalar x which maximizes $h(x, t)$. The subscript k shows the band number of the HSI image with B bands. It is empirically observed that the noise is higher in the VNIR region and therefore a bias towards lower band is created. Maximizing $h(x, t)$ captures the floor of the noise in VNIR which is consistent with most SWIR bands. This test is performed on different datasets and an empirical formula is established based on our observation.

Threshold \tilde{t} is found from (4.17), respectively. We applied this filter on our test dataset which reduces noise, as shown in Figure 4.8.

$$h(x, t) = \sum_{k=1}^B e^{-(x-t_k)} \quad (4.28a)$$

$$J(x, t) = \max_x \{h(x, t)\} \quad (4.28b)$$

The gradient-descent algorithm finds x . Figure 4.8 shows output after filtering. Intermediate maps are created after the filtering process.

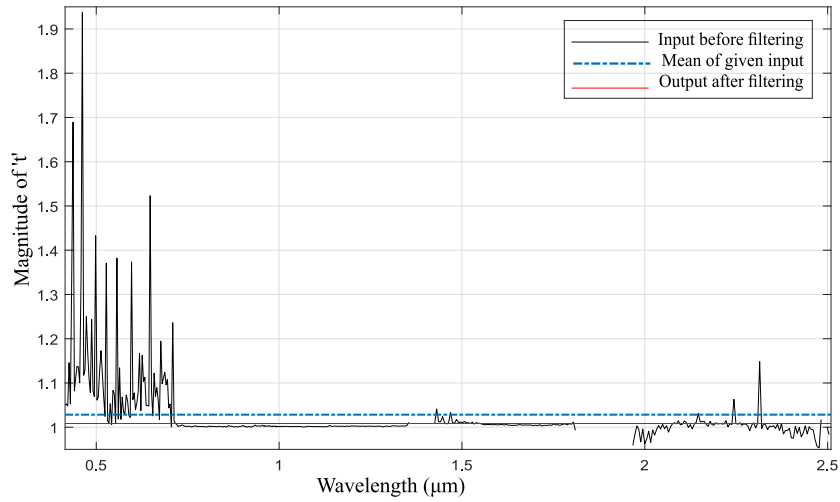


Figure 4.8: Effect of filtering on a given pixel.

4.3.4 Classification

The threshold map generated by Feature learning and filtering represents multiple levels of illumination. Therefore, we may generate a binary or multi-label classification from this intermediate map.

4.3.4.1 Binary Map

To create a binary map, our approach is similar to Equation (4.18) threshold function, shown in Equation (4.29) as:

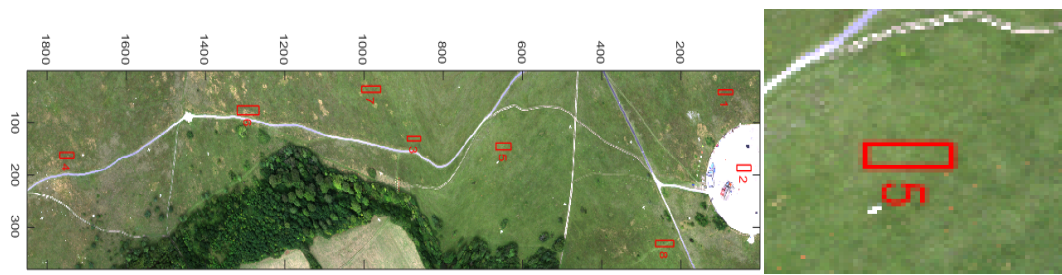
$$t = \begin{cases} > 1 & , \text{ (Non-shadowed)} \\ < 1 & , \text{ (Shadowed)} \end{cases} \quad (4.29)$$

Chapter 5

Experimental Data and Assessment Methodology

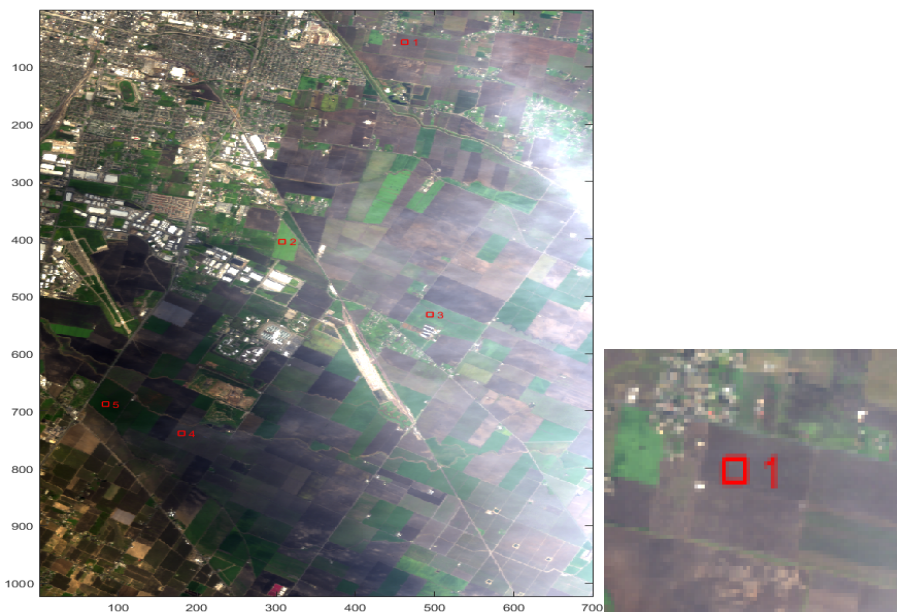
5.1 Experimental Data

For experimental validation, we have used two real images. Our algorithm is tested on the Selene SCIH23 dataset [10] which was acquired by the Defence Science and Technology Laboratory (DSTL) covering 0.4 to $2.5\mu\text{m}$. It is a registered image which is separately taken from HySpex VNIR-1600 (160 bands) and SWIR-384 (288 bands) sensors mounted on the same airborne platform. This scene was acquired near Salisbury, UK, on 12 August 2014 BST 13:00:04. The registered image has a GSD of 70×70 cm, QUAC was applied using ENVI software for atmospheric compensation. The second image is taken from Airborne Visible Infrared Imaging Spectrometer (AVIRIS) (224 bands), covering Modesto, California (Long $121^{\circ}17'45.9''$ N Lat $37^{\circ}56'44.37''$ W to $121^{\circ}4'50.31''$ N $37^{\circ}51'55.67''$ W) on 10 February 2015 BST 22:00:04. The latter is publicly available from <https://aviris.jpl.nasa.gov/>. RGB images of Selene and Modesto scenes are shown in Figure 5.1.



(a) Selene SCI H23 ($0.4 \mu\text{m}$ – $2.51 \mu\text{m}$, 448 bands), UK. The RGB image shows ROI marks used in experiments. ROI5 shows "Grass" patch in the field.

(b)



(c) AVIRIS Modesto ($0.36 \mu\text{m}$ – $2.49 \mu\text{m}$, 224 bands), California, USA. The RGB image shows ROI marks used in experiments. ROI1 shows "Farm Soil" patch in the field.

(d)

Figure 5.1: RGB color images of Selene and Modesto test scenes, respectively.

Figure 5.1 shows ROI selected in the Selene and Modesto scene to compute mean radiance of these selected regions.

5.1.1 Input Reflectance data

The input data for creating the classified texture is the reflectance (atmospherically compensated) image of the ground truth. Classification of the reflectance is performed on the whole cube containing all bands. A construction of simulated image from multi-spectral

or RGB image is out of the scope of this research. Some noisy bands in the reflectance data are removed to have realistic simulated radiance.

5.2 Experimental Setup for Atmospheric Parameters Search

The Selene scene is a vegetation biased scene, however, several fabric calibration panels with known reflectance are planted across the scene as shown in Figure 5.2. A white calibration panel with an average reflectivity of 99.0% and a black panel with average reflectivity of 2.0% are selected as reference materials. Moreover, there are seven large fabric panels with particular spectral properties were placed to act as calibration targets and deployed across the centre of the Hard Target (concrete) [10]. Figure 5.2 shows both ground (first row) and aerial (second row) image of the scene. Aerial image marks regions where some panels and tiles are planted.

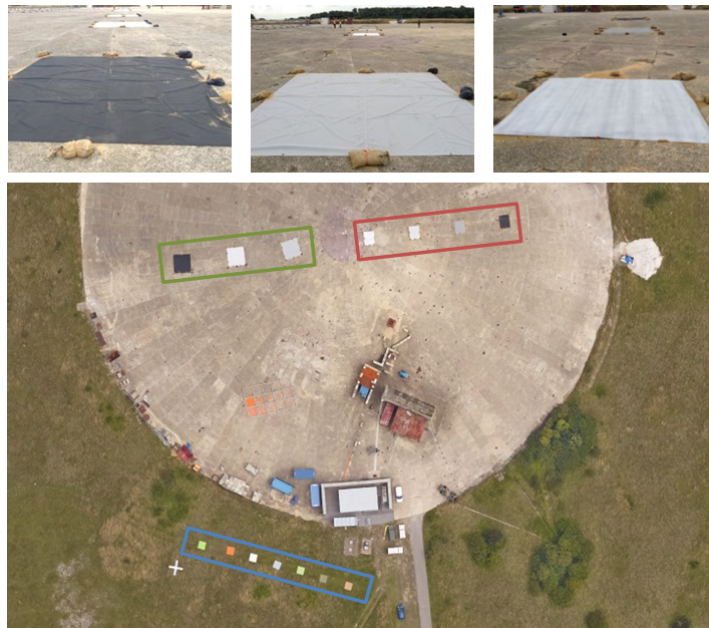


Figure 5.2: Commercially available fabric based calibration panels and tiles were planted in Selene scene [10]. Average reflectivity for calibration panels varies from 99.0% to 2.0%.



Figure 5.3: Location and reflectivity of calibration panels planted within the Ground Truth Selene scene [10].

Reflectivity of seven calibration panels planted on the hard target is given in Figure 5.3. The primary purpose of planting these panels is to facilitate atmospheric compensation. ELM, for example, requires to have in-scene calibration panels for estimating reflectance. In Selene test, high reflectance Acrylic sheet also known as perspex are used. It is available in white, green and orange colours. These sheets are planted to analyse the performance of detection for several different sizes. More details about the calibration panels and intended experiments are provided in [10].

5.3 Validation Data for Shadow Detection

The Selene SCIH23 scene's terrain was mapped with a high-resolution LIDAR to create a contour-based DSM of the scene. This DSM is used in a ray-tracer to generate a shadow map, which creates ground truth for Selene results. Modesto does not have DSM data available. Therefore, classification performance is tested on visual perception only.

5.4 Validation of Simulated Images

There are several approaches to validate a simulated image in literature. [86],[87] employed computing eigenvectors of the covariance matrices of simulated and real scenes to establish a similarity metric between the scenes. Moreover, [86] applied K-means classification with fewer (≤ 5) classes to generate a classified pattern in the real and simulated scene to compute a confusion matrix for quantifying similarity. [88] compared CameoSim/ Radtherm generated images with their respective ground truths by linear discriminant analysis using Bhattacharya distance as metrics [89, 90]. Selex Galileo's simulation image was validated by clay soil mapping at airborne scale and its down-scaling (lower resolution) at satellite level [91]. DIRSIG 3 validation by [92] used the band to band histogram comparison of reference and simulated images and by detection performance (ROC curves) of the two images computed by ACE. For validation of DIRSIG 4, Gaussian Maximum Likelihood and target detection by RX anomaly detector were employed [93]

5.4.1 Our Validation Approach

On the basis of the reported validation approaches we have selected following metrics measures for validation.

- Eigenvectors of covariance matrix of the whole scene
- ℓ_1 -norm error
- Normalized Root Mean Square Error (NRMSE)
- ROC curve for targets with and without classification

5.4.1.1 Eigenvectors of covariance matrix of the whole scene

Several target detection algorithms such as Adaptive Cosine Estimator (ACE), Signature Matched Filter (SMF), Generalized Likelihood Ratio Test (GLRT), RX Anomaly detector depends on in-scene covariance. To achieve detection performance of these algorithms similar to ground truth, it is important that band to band covariance of a simulated scene matches the ground truth covariance. This similarity measure is quantified by computing

the first four Eigenvectors of the GT and simulated scenes and computing NRMSE error of the simulation with respect to GT.

We have compared the first four Eigenvectors to evaluate the similarity of covariance statistics. A similar approach is also used by [86] in their experiments with DIRSIG4. In their experiments the first two eigenvectors show significant matching while further vectors are shown to capture noise.

Band to band covariance of the whole scene of dimension $b \times b$ is computed, where b is the number of bands. First four Eigenvectors are calculated for the simulated scene of CHIMES and CameoSim and a comparison of vectors is performed by means of NRMSE error.

5.4.1.2 ℓ_1 -norm Error

Generally, ℓ_p -norm of a vector is given as;

$$\|x\|_p = \left(\sum_i |x_i|^p \right)^{\frac{1}{p}}. \quad (5.1)$$

ℓ_1 -norm error, therefore, is the absolute difference between the two entities. ℓ_1 -norm is used in literature for HSI scene simulation where sensor characteristics are varied such as [94], [95]. In the former the magnitude variability between test and reference radiance is of the same order, in the latter, a template-based matching is performed for the reference signature in the scene.

In at-sensor radiance estimation problems, large variations between GT and simulations are typically due to mismatch in atmospheric data generated by user input and the GT. As CHIMES incorporate automation of this search, therefore the magnitude of difference between GT and the simulated scene is assumed to be lower. This motivated us to employ ℓ_1 -norm error as it inherently reports the performance of our atmospheric search module.

ℓ_1 -norm error is calculated for the whole scene to show the extent of error and error pattern. ℓ_1 -norm error is calculated by Equation (5.2), where B is the number of spectral bands.

$$\ell_1\text{-norm}(GT\text{Radiance}, Simulated\text{Radiance}) = \sum_{n=1}^B |GT\text{Radiance} - Simulated\text{Radiance}| \quad (5.2)$$

5.4.1.3 Normalized Root Mean Square Error

Goodness of Fit is calculated by means of NRMSE in atmospheric search algorithm. Moreover NRMSE is also calculated for mean radiance of different material ROIs. NRMSE is calculated by Equation (5.3) for a pixel and Equation (5.4) for an ROI. In Equation (5.3) only one pixel is compared between simulated and ground truth images. Mean GT Radiance of the whole scene is calculated which is biased towards most abundant material. In the case of NRMSE of ROIs mean radiance of ROI is calculated instead of a single-pixel as in the previous case.

$$NRMSE_{Pixel} = 1 - \frac{(\ell_2\text{-norm}(GT\text{RadiancePixel}, SimulatedRadiancePixel))}{\ell_2\text{-norm}(GT\text{RadiancePixel}, MeanGT\text{RadianceWholeScene})} \quad (5.3)$$

$$NRMSE_{ROI} = 1 - \frac{(\ell_2\text{-norm}(MeanGT\text{RadianceROI}, MeanSimulatedRadianceROI))}{\ell_2\text{-norm}(MeanGT\text{RadianceROI}, MeanGT\text{RadianceWholeScene})} \quad (5.4)$$

We mentioned a potential issue with the comparison of radiance in the previous section. In order to overcome the magnitude difference of radiance in ℓ_1 -norm error, NRMSE error is also employed in the result validation process.

5.4.1.4 Alternative Methods

A number of performance evaluation alternatives exist for evaluating simulated images. These include the unsupervised classification of both GT and Simulated images and comparing the similarity between the results [86]. Although classification based approaches typically create a pattern shown by colours and provide a visually apparent similarity, however quantification may require some user interaction. Therefore, this approach was not chosen for our analysis. Another approach is to perform atmospheric compensation on the radiance image and detecting an in-scene target. ROC curves comparison with ground truth is a performance metric in this case. Such techniques are used in [86],[91]. This method incorporates artefact of atmospheric compensation algorithms. Moreover, classified textures typically reduce the performance of detection due to classification required for efficient rendering. Therefore, validation of simulated images by detection was not taken into account. The second-order statistical metric, Kurtosis is also applied in liter-

ature [96],[97] for comparison between reconstructed HSI and atmospherically compensated GT reflectance data. It is however not reported for evaluation of simulated radiance in the literature.

Chapter 6

Results and Validation

6.1 Results and Validation

In this section, we will present results of the Selene and Modesto scenes. Construction of GT for the Selene scene enables us to provide a more quantitative validation by means of confusion matrix, while for Modesto, it is more qualitative and evaluated visually. For comparative analysis, we have considered exploiting descriptor-based shadow detection from RGB input, as in Gevers [84], and Beril [79] (see [98] for source code), and methods that take in multispectral radiance input, as in RGBN [82] requiring RGB and a single NIR band, and false-color shadow detection method, LULC [81] (see [99] for source code), requiring five input bands within $0.3 \mu\text{m}$ to $2.5 \mu\text{m}$.

6.2 Simulation Results

Simulation results are assessed based on variable terrain types, atmosphere types and adjacency models as discussed in Section 3.5.2. Summarized results for each of these parameters are included subsequently, however, more detailed material-wise statistical analysis is included in Appendix A. Error metrics used for evaluation are defined in Appendix 5.4.1.2.

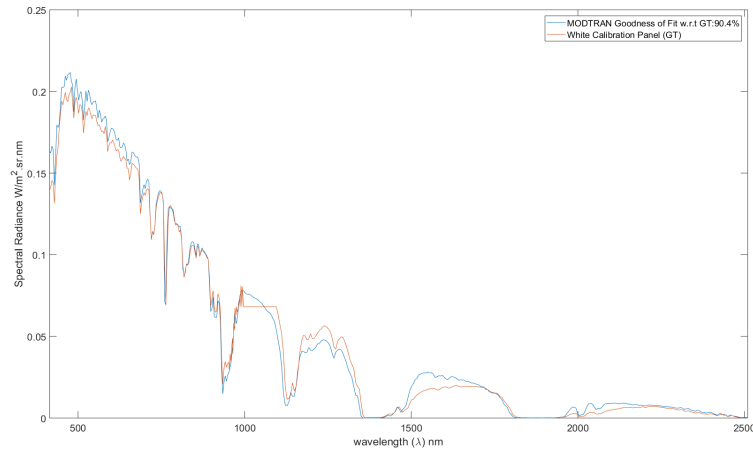
We report the results of the search algorithm for atmospheric parameterization in the subsequent section, followed by results of simulations. In atmosphere search we observe a closer match between white panel's radiance and MODTRAN's L_w , however relatively larger error for MODTRAN's path radiance L_b and black panel's radiance. In simulation results, we observe a closer match between CHIMES BOAEM and CameoSim under cloudy atmosphere and flat terrain, in terms of radiance error ranges and patterns. Moreover, the material wise results in Appendix A also shows a closer agreement between

CHIMES (BOAEM) and CameoSim. It is shown in the appendix that TIAEM produces better spectral reflectance shapes (colours) for the man-made target materials, particularly in the VNIR region. In the case of a clear sky, all models produce similar results.

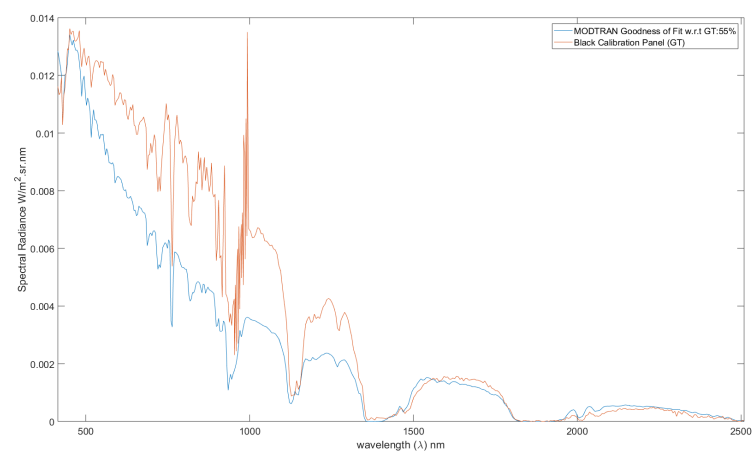
6.2.1 Automatic Search for Atmospheric Parameters

During the search process, MODTRAN is input with several parameters, few of them are shown in Table 3.1. In our experimental data, sensor altitude is higher than cloud base and presence of thin cloud below the sensor is also reported [10], therefore we will vary the aerosol optical thickness below the sensor to search the approximate underlying atmospheric condition in the scene.

In each iteration MODTRAN's L_w that is, accumulated total ground reflected radiance and path radiance is calculated and goodness of fit is recorded. Estimate of searched white panel radiance L_w and black panel radiance L_b is shown in Figure 6.1, that possesses the highest goodness of fit, that is 90.4% of match in terms of NRMSE for L_w and 55.26% for L_b , respectively. These results manifest good atmospheric search in terms of total ground reflected radiance, however, estimation of path radiance is relatively worse. If path radiance estimated in the search process is lower than the GT panel then it may result in higher contrast than the GT. As the search method only employs MODTRAN at-sensor radiance estimate, the effect of volumetric scattering is not accounted for. At rendering time, volumetric scattering may either reduce the error in the estimate of L_b or increase it depending on the choice of phase function and backscattering parameter, such as the variable g in Henyey-Greenstein function [48]. However, due to good estimation of the TGRR, we have selected searched parameters ($C_w = 0.22 \text{ gm/m}^3$ and $R = 0.024 \text{ Km}$) and fixed parameters given in Table 3.1 for our overcast condition simulations. In the case of atmospheric search by using white and black panel, it is easier to segregate the path radiance and TGRR, therefore these targets were employed. A similar search based on orange perspex yielded ($C_w = 0.22 \text{ gm/m}^3$ and $R = 0.023 \text{ Km}$), which is very close to white and black panel's parameters.



(a) White Panel Estimation



(b) Black Panel Estimation

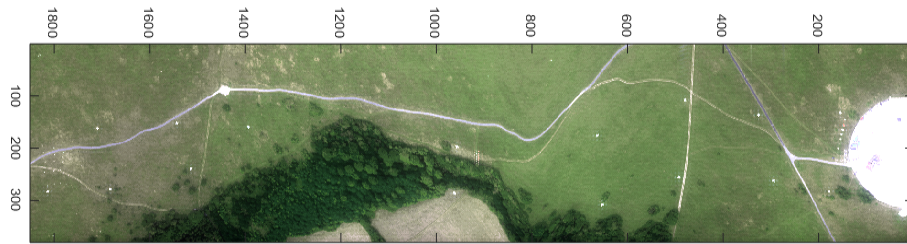
Figure 6.1: White and black panel estimates from atmospheric search.

6.2.2 CHIMES and CameoSim Simulation Results with Flat Terrain

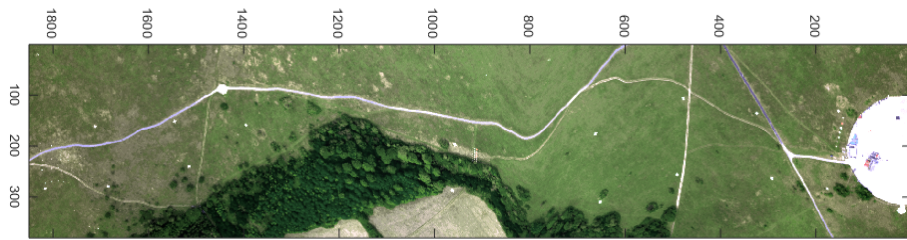
RGB images of Selene scene rendered by CHIMES and CameoSim are shown in Figure 6.2. Inputs and parameters to both simulators are the same, particularly the atmospheric condition is cloudy with parameters shown in Table 3.1 and the scene geometry is flat terrain. We observe illumination differences in the RGB images, particularly at the region having trees, closer to the farm. It should be noted that the region with trees and buildings on the concrete hard target show a higher error. The scene possess both trough and crest bumps which cause a higher error. Another contributing factor that accumulates the error, is the presence of shadows [85]. As shown in Figure 6.3 (b,c), CHIMES

BOAEM and CameoSim, apparently have closer ℓ_1 -norm error, the error map also shows a closer error range and mean error, compared to TIAEM. The error range of BOAEM is between 0–3.16 with a mean error of 1.05. CameoSim error lies in the range of 0–3.74 with a mean error of 1.25. The visual pattern of the error map does look similar to discrepancies in the tree region in CHIMES BOAEM which has high error relative to CameoSim. On the other hand CHIMES TIAEM error map has a higher range, lying between 0–3.99 with a mean error of 1.33.

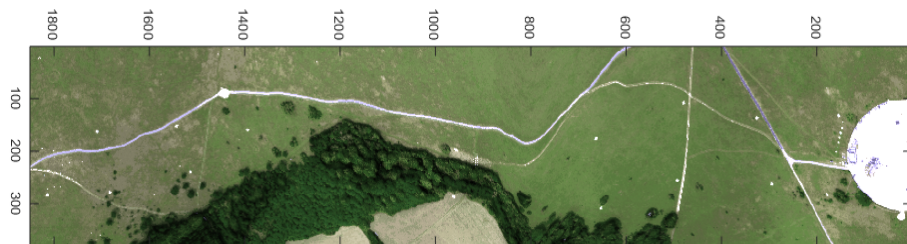
All three error maps have higher error in tree, building and low reflectance target regions. When a region of an HSI image has low intensity it is either due to low reflectance target or shadow. Low intensity implies low radiance, therefore atmospheric compensation typically yields noisy reflectance for that region. It is also noticed that some material reflectance does not provide the information required for good detection, experimental data about the error in low radiance region can be found in [100]. The region of Selene scene with forest/tree have fragmented regions between adjacent trees, therefore this region has a larger contribution for shadows. Similarly, building on hard target also cast shadows. This explains the cause of higher error in these regions.



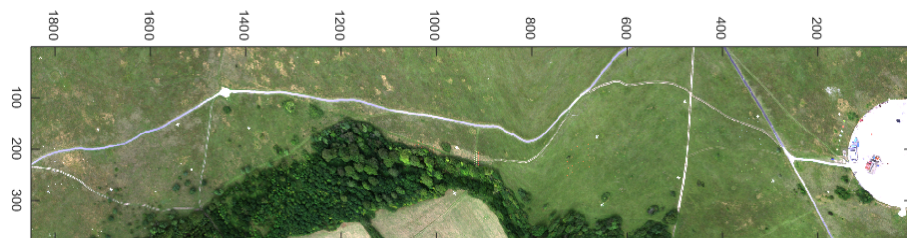
(a) CHIMES TIAEM simulated Imagery



(b) CHIMES BOAEM simulated Imagery



(c) CameoSim simulated Imagery



(d) Ground Truth Imagery

Figure 6.2: RGB color image for simulated and Ground Truth Selene scene. Atmospheric condition listed in Table 3.1.

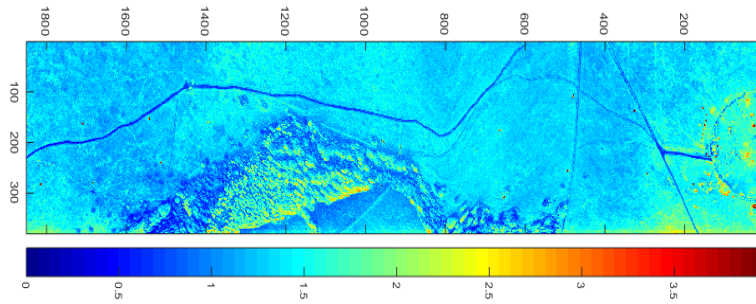
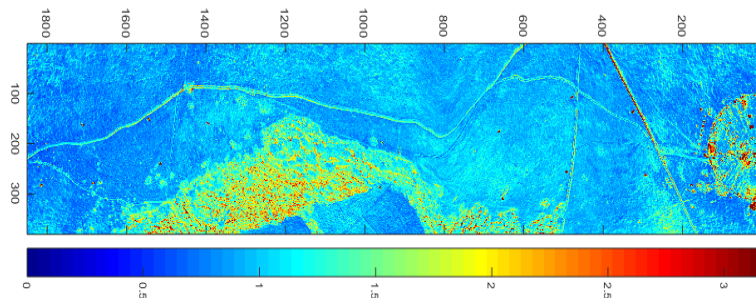
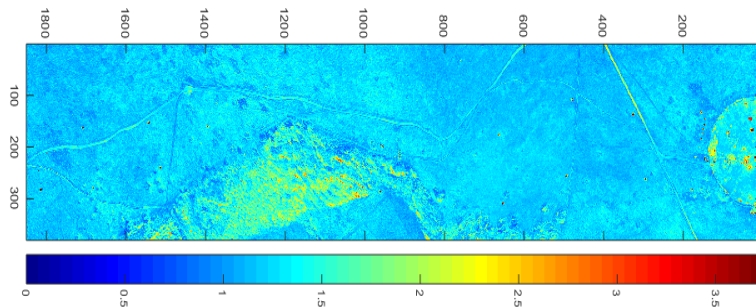
(a) CHIMES TIAEM ℓ_1 -norm error range [0 3.99], Mean error [1.33](b) CHIMES BOAEM ℓ_1 -norm error range [0 3.16], Mean error [1.05](c) CameoSim ℓ_1 -norm error range [0 3.74], Mean error [1.25]

Figure 6.3: ℓ_1 Norm error of simulated scenes with respect to Ground Truth in atmospheric condition of Table 3.1.

Several ROIs are considered for evaluating the mean of radiance across the scene, these ROIs are shown in Figure 5.1. Selection of these ROIs is biased towards vegetation which constitutes more than 80% of the scene. As mentioned in Section 6.2.2, the rationale behind this ROI selection is in the evaluation of background statistic of models. Out of this eight, seven are grass, while one is from the concrete hard target. Mean spectral radiance of these ROIs is shown in Figure 6.4. NRMSE and ℓ_1 -norm errors are tabulated in Table 6.1, which shows that CHIMES BOAEM has consistently lower errors compared

to TIAEM and CameoSim.

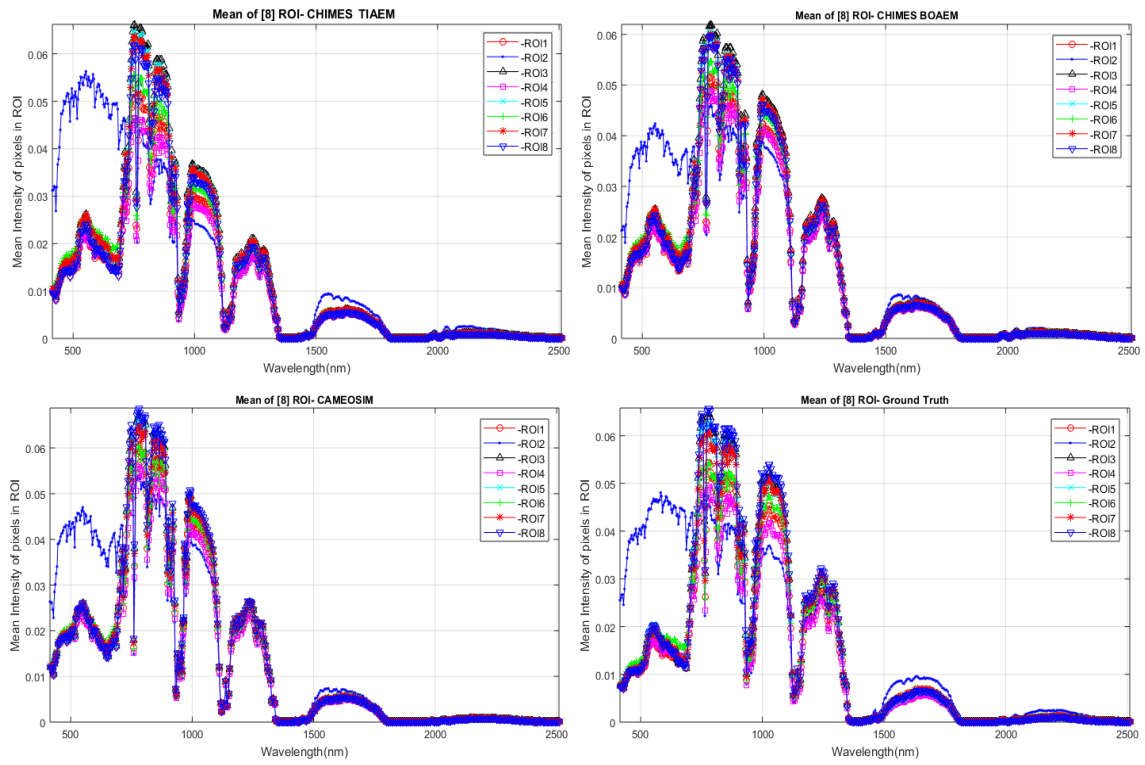


Figure 6.4: Mean radiance of ROIs shown in Figure 5.1.

Table 6.1: NRMSE, ℓ_1 -norm error of radiance in Figure 6.4. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.

ROI	Error	CHIMES TIAEM	CHIMES BOAEM	CameoSim
ROI 1	NRMSE	29.92%	17.60%	24.82%
	ℓ_1 -norm	1.37	0.92	1.21
ROI 2	NRMSE	30.18%	19.45%	22.87%
	ℓ_1 -norm	1.63	1.10	1.270
ROI 3	NRMSE	25.97%	17.25%	22.05%
	ℓ_1 -norm	1.38	1.01	1.234
ROI 4	NRMSE	29.43%	16.73%	26.69%
	ℓ_1 -norm	1.25	0.77	1.163
ROI 5	NRMSE	25.47%	16.90%	21.67%
	ℓ_1 -norm	1.32	0.95	1.172
ROI 6	NRMSE	29.90%	17.57%	24.41%
	ℓ_1 -norm	1.25	0.93	1.25
ROI 7	NRMSE	27.54%	17.48%	23.10%
	ℓ_1 -norm	1.42	1.01	1.26
ROI 8	NRMSE	31.91%	20.77%	22.313%
	ℓ_1 -norm	1.67	1.21	1.27

Similarly, man made material radiance shown in Appendix A also show closer match between BOAEM and CameoSim in terms of NRMSE error of mean radiance.

Eigenanalysis is also performed for evaluation of results. Eigenvectors of band to band covariance matrix are calculated. A NRMSE error between first four eigenvectors with respect to GT is shown in Figure 6.5. As the trend developed in a previous error metric, both BOAEM and CameoSim have similar NRMSE error magnitude. It should be noted that TIAEM has substantially lower error magnitude in first eigenvector. This trend also apparent in material wise eigenanalysis shown in Appendix A.

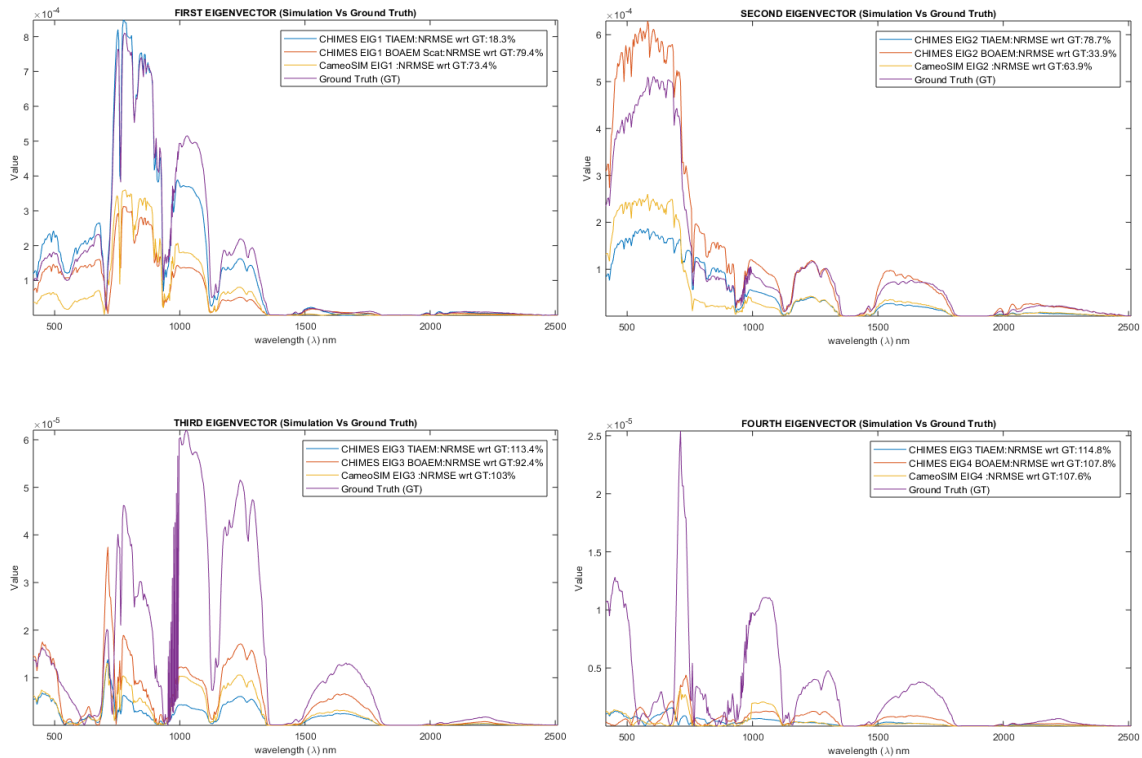


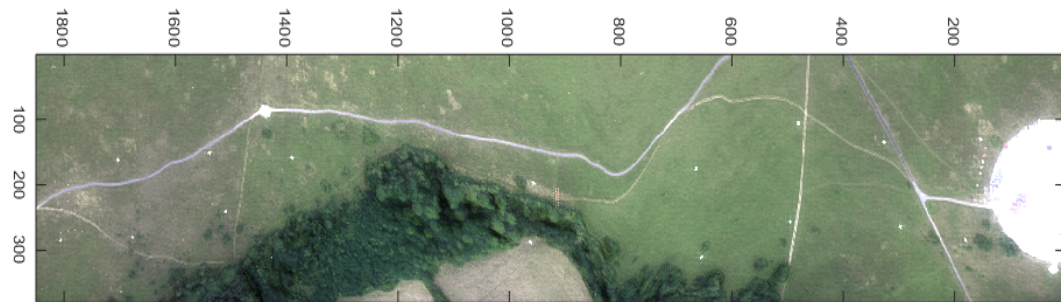
Figure 6.5: Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GT's.

For example, TIAEM has 18.3% NRMSE compare to 79.4% and 73.4% of BOAEM and CameoSim, respectively, for first eigenvector. In the next two eigenvectors BOAEM has the lowest error magnitude, while CameoSim has the lowest error in the last one. In case of material wise radiance analysis presented in Appendix A, CHIMES TIAEM and BOAEM have the lower NRMSE error with respect to GT compared to CameoSim. For example in orange perspex near grass TIAEM has an error of 13.2%, while CHIMES BOAEM has an error of 8.8%, compared to CameoSim's 89.9%. For concrete, TIAEM has the error magnitude of 28.0% compared to BOAEM's 26.8% and CameoSim's 25.5%. Therefore CameoSim performs slightly better for concrete, in terms of eigenvector matching with GT.

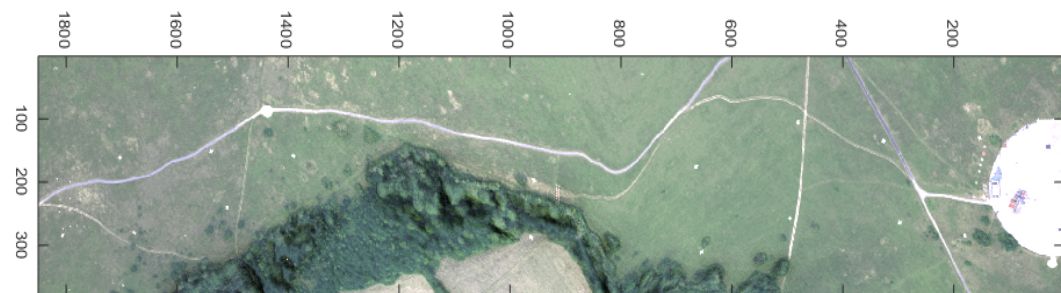
Figures A.2 and A.4 clearly show influence of albedo material that is, grass scrub in VNIR region in BOAEM and CS results. However, TIAEM does not depict this over influence of the average background albedo material used in MODTRAN calculations. Even though TIAEM still cause increase in radiance at high albedo bands. Similarly this trend is visible in the DEM based results as well.

6.2.3 CHIMES and CameoSim Simulation Results with DEM

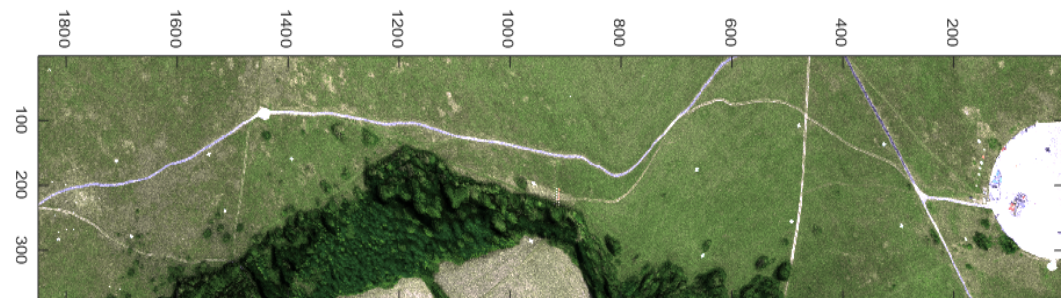
Rugged terrain causes a non unity sky-view factor, which attributes to inclusion of background reflection and reduced diffuse radiance. It is therefore important to validate the differences in radiance with and without terrain. RGB images of simulated scene with DEM surface is shown in Figure 6.6. In case of RGB channels, we observe that in contrary to results of flat terrain, the contrast in both models of CHIMES images has substantially reduced, particularly in the region where terrain is higher. This manifests the background contribution due to reflection. In CameoSim image we observe reduced spatial reconstruction quality leading to apparent pixelation. However, the visual contrast appears to increase in CameoSim compared to the flat terrain simulation. A more clearer picture about error is found in ℓ_1 -norm error map in Figure 6.7, subsequently.



(a) CHIMES TIAEM simulated Imagery



(b) CHIMES BOAEM simulated Imagery



(c) CameoSim simulated Imagery

Figure 6.6: RGB color image for simulated and Ground Truth Selene scene. Atmospheric condition is given in Table 3.1.

Figure 6.7 shows that CHIMES TIAEM has an error range of 0–4.13, with the mean error of 1.38 compared to 0–3.99, with the mean error of 1.33, in case of flat terrain. The increase in mean error is therefore 3.6 %. Similarly, CHIMES BOAEM has an error range of 0–3.28, with the mean error of 1.09 compared to 0–3.16, with the mean error of 1.05, in case of flat terrain. The increase in mean error in this case is 3.7%. For CameoSim result, the error range is 0–18.33, with the mean error of 6.11 compared to 0–3.74, with the mean error of 1.25 for flat terrain. The increase in error is 388.8%. The relative pattern of error as shown Figure 6.7 is similar to that in flat terrain case.

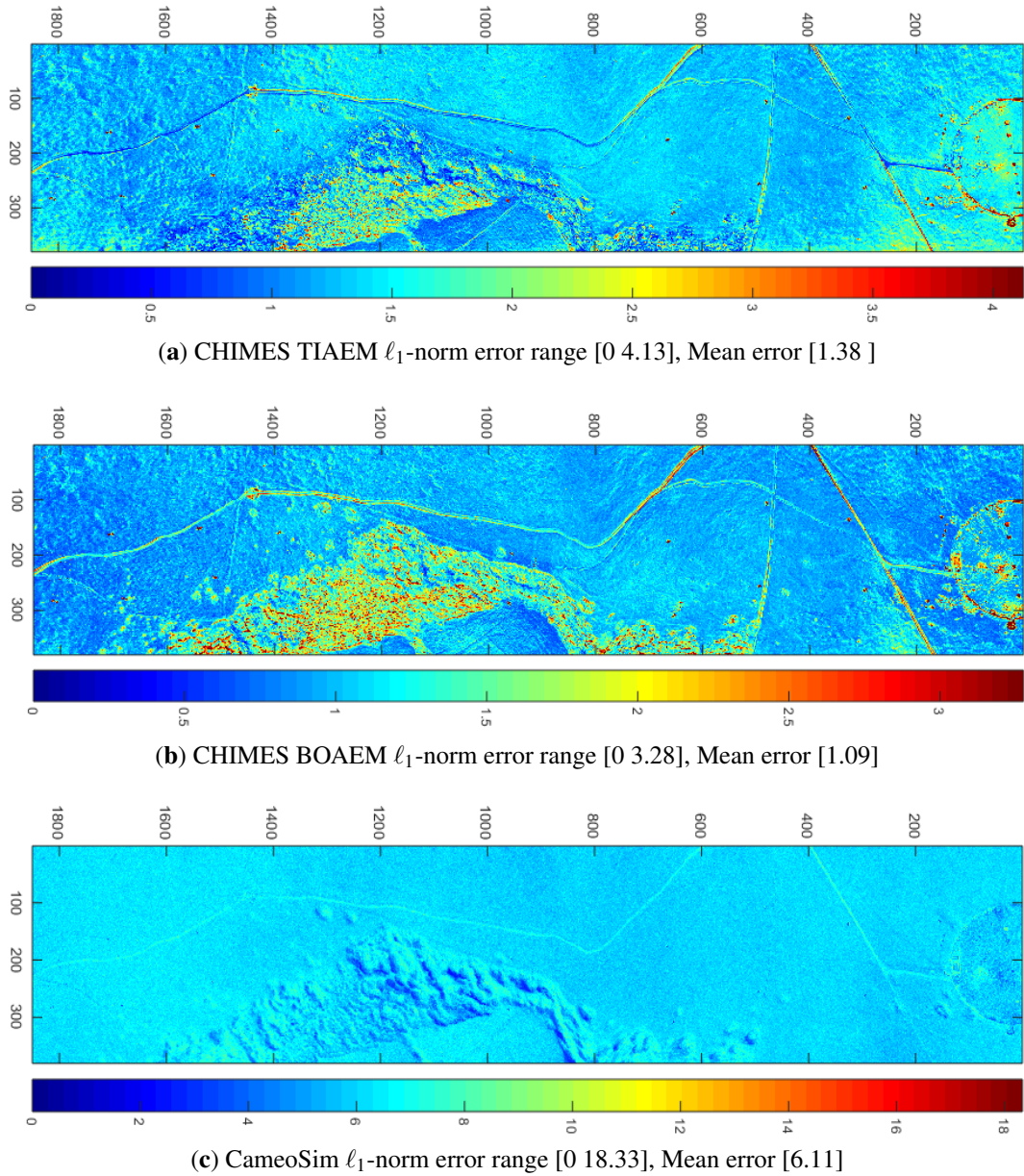


Figure 6.7: ℓ_1 Norm error of simulated scenes with respect to Ground Truth.

When a realistic and LIDAR based DEM map is used to construct a terrain, a natural expectation was to see a reduction in the mean error. However, it is instead noticed that both simulators have shown an increase in the mean error compared to the flat terrain. The extent of increased error in CameoSim may not be explained as we do not have access to their underlying model and source code. However, in the case of CHIMES, the primary cause lies in the drawback of automated atmospheric search process which is described in

section 3.6.2.1. The search process runs MODTRAN in a number of iterations where the geometry/terrain is only allowed to be flat, due to the limitation in MODTRAN. Therefore the best match of reference material radiance and in-scene radiance of that material does not incorporate the DEM effect. Contrarily, it works better in the case of flat terrain due to its support and utilization in MODTRAN's iterations.

We present the ROI based mean radiance magnitude like presented in flat terrain case. The radiance is increased in all simulations due to background reflected contributions. Figure 6.8 and Table 6.2 show that CHIMES BOAEM has the least NRMSE and ℓ_1 Norm error in almost all ROIs. It is encouraging to note that CHIMES BOAEM's error reduced for all eight ROIs when DEM is introduced. CHIMES TIAEM however maintains similar error as in flat terrain case. The CameoSim result for ROIs shows an enormous increase in errors.

Similar to flat terrain, eigenvectors for the DEM are also calculated. Eigenvectors' plot is shown in Figure 6.9. The NRMSE error in this case observe an increase in simulated results. For example, TIAEM has 44.2% NRMSE compared to 72.6% and 53.4% of BOAEM and CameoSim, respectively, for the first eigenvector. Similar to flat terrain, in the second two eigenvectors BOAEM has the lowest error magnitude. The error magnitude of all simulation results exceed 100% in third and fourth eigenvector cases.

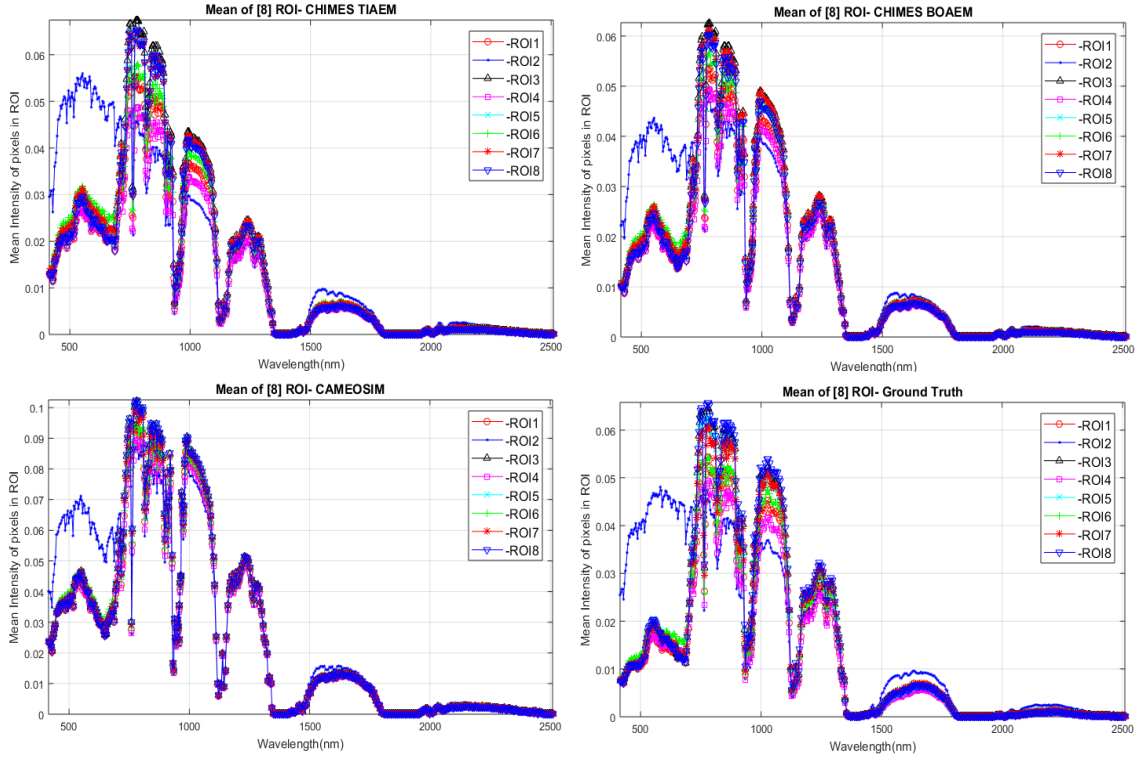


Figure 6.8: Mean radiance of ROIs shown in Figure 5.1

Table 6.2: NRMSE, ℓ_1 -norm error of radiance in Figure 6.8. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.

ROI	Error	CHIMES TIAEM	CHIMES BOAEM	CameoSim
ROI 1	NRMSE	27.51%	16.98%	118.30%
	ℓ_1 -norm	1.33	0.91	6.16
ROI 2	NRMSE	29.79%	17.80%	112.32%
	ℓ_1 -norm	1.65	0.97	5.99
ROI 3	NRMSE	25.04%	16.95%	96.66%
	ℓ_1 -norm	1.36	1.01	5.90
ROI 4	NRMSE	28.45%	17.23%	134.02%
	ℓ_1 -norm	1.24	0.81	6.34
ROI 5	NRMSE	24.89%	16.62%	99.94%
	ℓ_1 -norm	1.31	0.94	5.95
ROI 6	NRMSE	28.25%	18.35%	116.07%
	ℓ_1 -norm	1.43	1.06	6.09
ROI 7	NRMSE	26.09%	17.25%	101.74%
	ℓ_1 -norm	1.42	1.01	5.91
ROI 8	NRMSE	30.92%	19.87%	91.24%
	ℓ_1 -norm	1.66	1.18	5.72

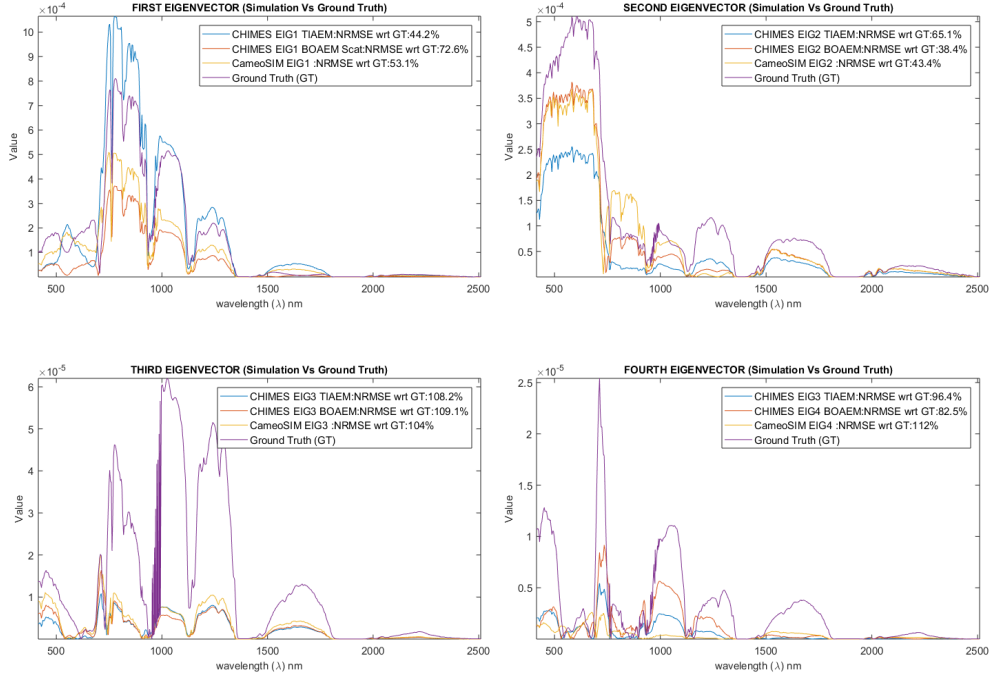


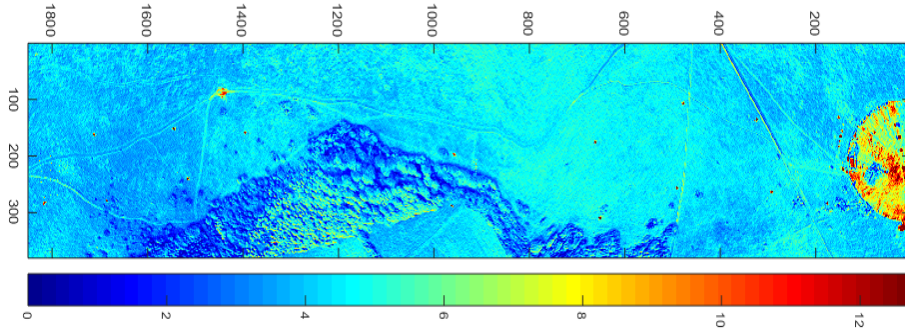
Figure 6.9: Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GTs.

6.2.4 CHIMES and CameoSim Simulation Results with Clearsky

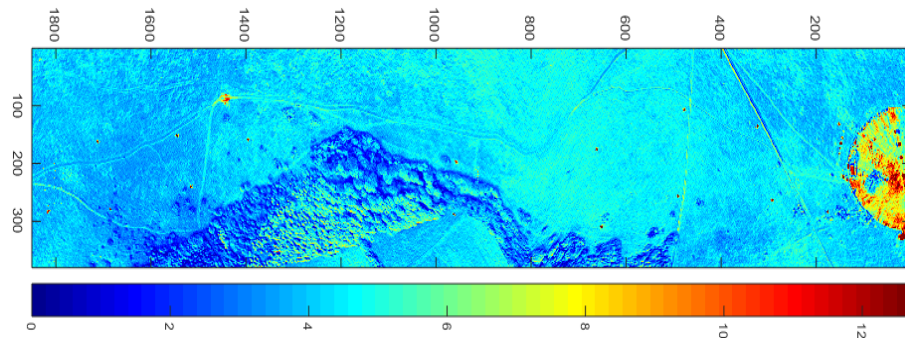
We have briefly included the results of a simulation in a clearsky atmosphere. There are no aerosols and clouds in these simulations. This is a scenario where scattering has little affect therefore adjacency effect should not be observed. Lack of aerosol increases transmission and therefore the at-sensor radiance magnitude is higher. We include ℓ_1 -norm error map and radiance of white panel to show a comparison of simulations.

Due to the lack of adjacency effect, a closer match between CHIMES TIAEM and BOAEM models is expected. It is premised by the ℓ_1 -norm error map shown in Figure 6.10. Although we calculate ℓ_1 -norm error with respect to ground truth, however as the atmosphere is far from the true scene's atmosphere, therefore error magnitude have no significance. GT is only used as a reference radiance to show similarity between the simulation models. In ℓ_1 -norm error CHIMES TIAEM has an error range of 0–12.79, with the mean error of 4.28. CHIMES BOAEM has an error range of 0–12.70, with the mean error of 4.23. CameoSim's error is in range 0–14.71, with the mean error of 4.90, as shown Figure 6.10. We observe a close match in error ranges, mean error and error pattern across the scene.

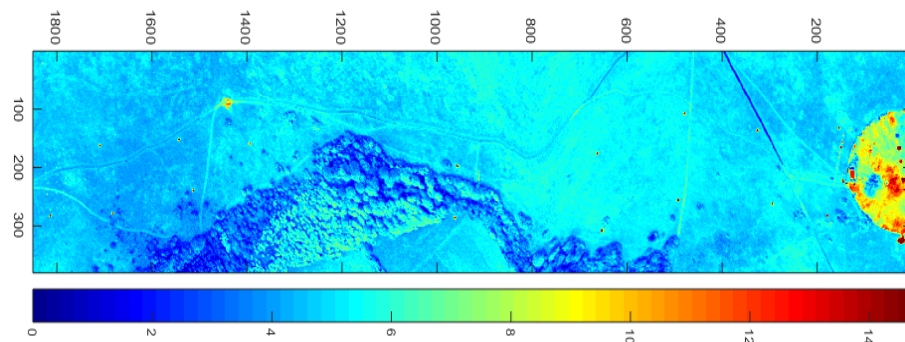
Radiance from ROI of white panel is shown in Figure 6.11. Both magnitude and signature of output radiance for all simulation models look in close agreement with each other.



(a) CHIMES TIAEM ℓ_1 -norm error range [0 12.79], Mean error [4.28]



(b) CHIMES BOAEM ℓ_1 -norm error range [0 12.70], Mean error [4.23]



(c) CameoSim ℓ_1 -norm error range [0 14.71], Mean error [4.90]

Figure 6.10: ℓ_1 -norm error of simulated scenes with respect to Ground Truth in clear sky atmosphere. GT radiance is only used as reference for calculating error.

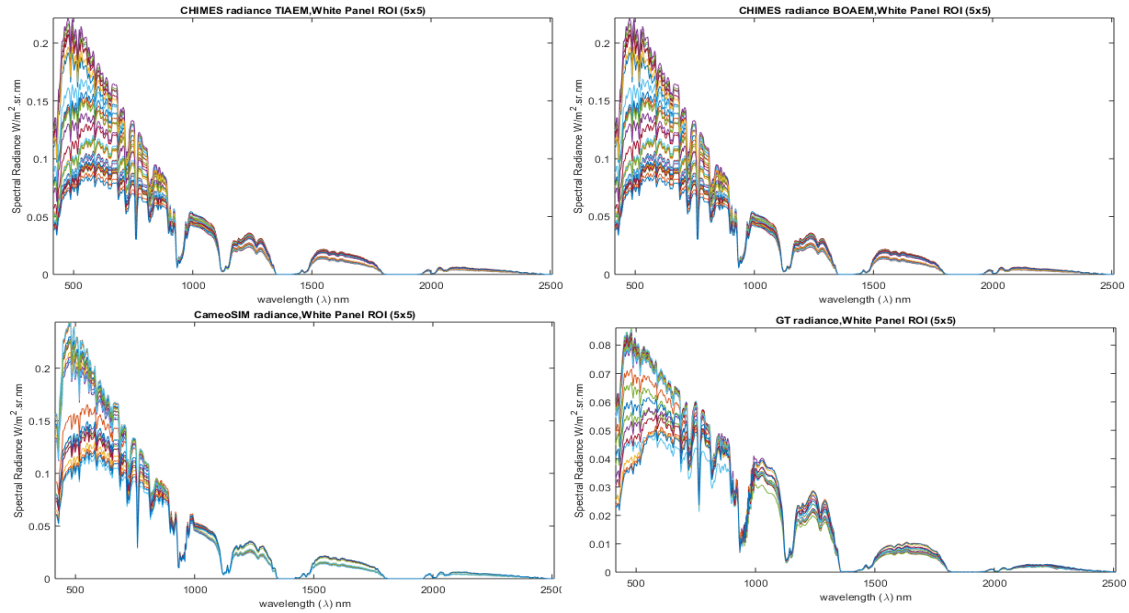


Figure 6.11: Radiance of white calibration panel in a (5×5) ROI. Simulated radiance are for clearsky while ground truth (GT) is under cloudy condition. GT is only used as a reference in this comparison.

6.3 Modesto Simulation

A distinctive feature of Modesto scene is the sensor altitude, which is a key factor in our localized adjacency model. Moreover the scene image is partly covered with clouds and the shadow casted by the cloud is apparant. It has vegetation, soil and urban region across the swath. The ℓ_1 -norm error map of TIAEM, BOAEM and CameoSim image shows that TIAEM has the lowest mean error of 0.81, compared to BOAEM's 0.86 and CameoSim's 0.9.

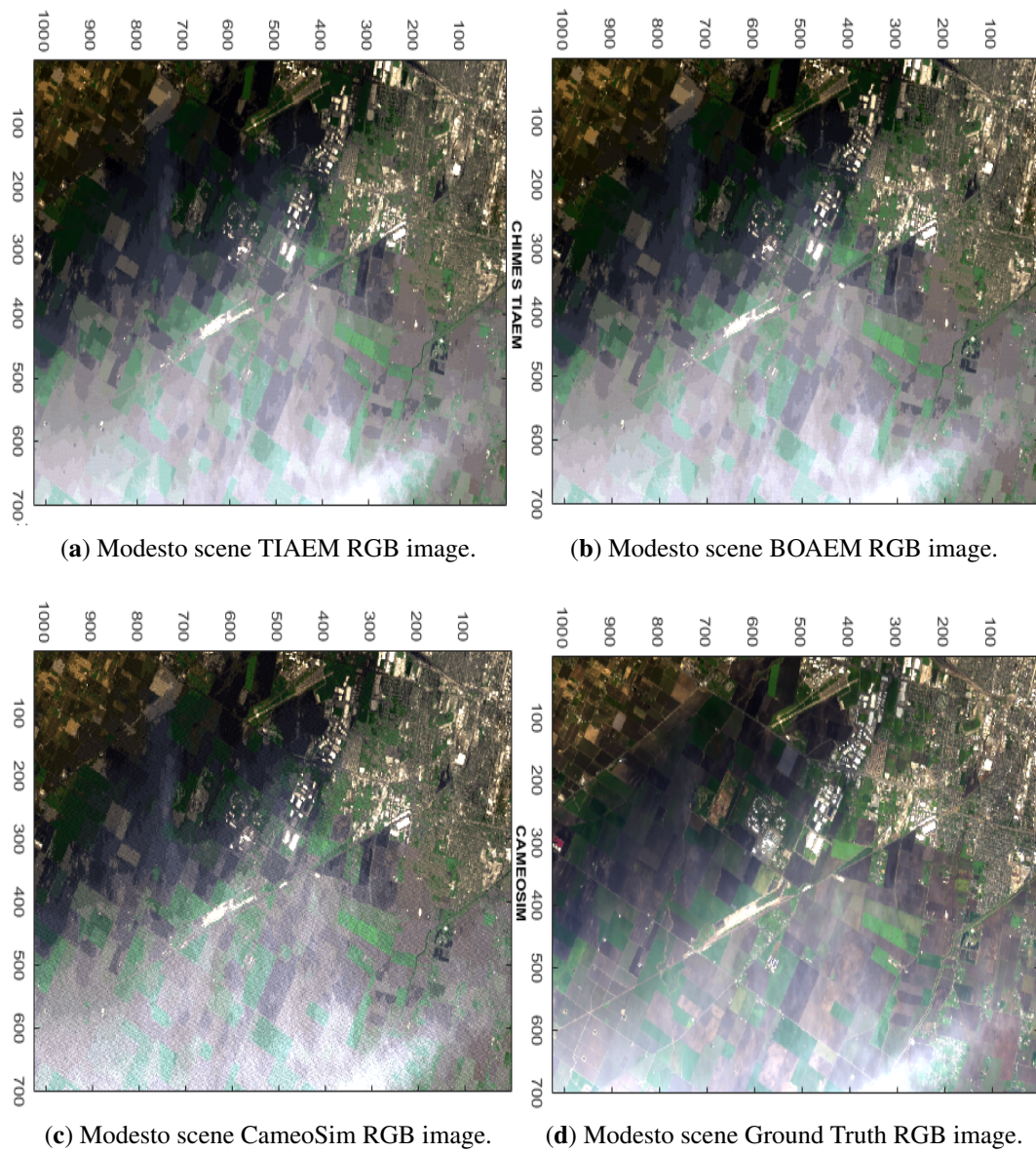


Figure 6.12: RGB image from Simulation of Modesto scene with CHIMES, CameoSim with Ground Truth.

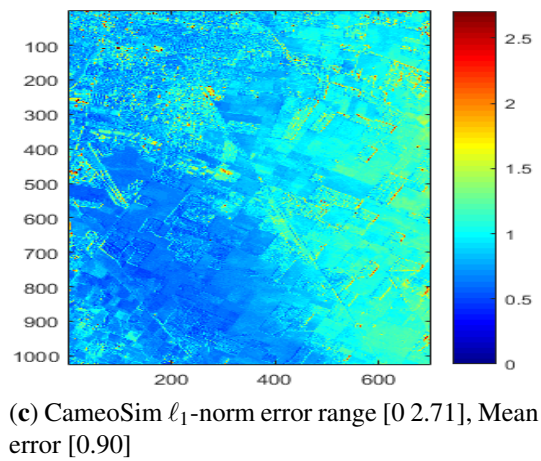
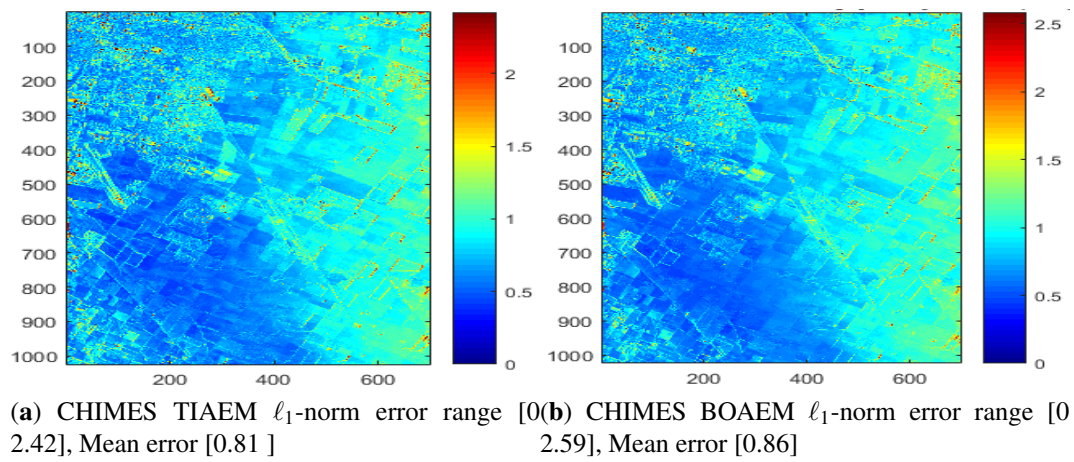


Figure 6.13: ℓ_1 Norm error of simulated scenes with respect to Ground Truth.

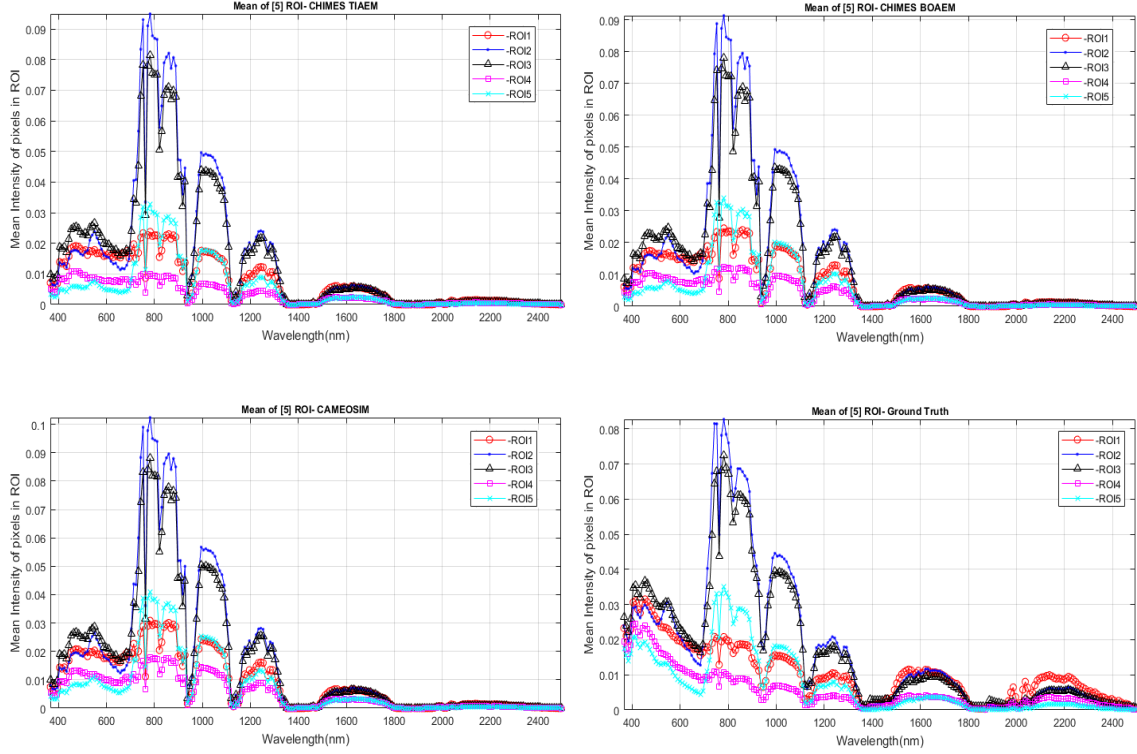


Figure 6.14: Mean radiance of ROIs shown in Figure 5.1

Table 6.3: NRMSE, ℓ_1 -norm error of radiance in Figure 6.8. Material of ROI 2 is concrete, other ROIs are grass across the scene. Least errors are shown in bold text.

ROI	Error	CHIMES TIAEM	CHIMES BOAEM	CameoSim
ROI 1	NRMSE	61.66%	73.498%	75.50%
	ℓ_1 -norm	29.87	30.587	14.76
ROI 2	NRMSE	31.41%	31.716%	37.21%
	ℓ_1 -norm	13.01	15.829	16.34
ROI 3	NRMSE	31.45%	33.054%	37.45%
	ℓ_1 -norm	13.14	16.880	15.57
ROI 4	NRMSE	69.66%	73.557%	76.21%
	ℓ_1 -norm	40.00	31.582	32.38
ROI 5	NRMSE	44.69%	45.035%	47.15%
	ℓ_1 -norm	15.72	20.595	17.42

In Modesto scene, we observe higher NRMSE error for all ROIs, which persists in both BOAEM and TIAEM models of CHIMES and also in CameoSim. All three models show a larger deviation in the visible range. This shows classification and material allocation error in the process of constructing the classified texture. The number of classes for

K-mean clustering was set to 120 as in Selene scene. Modesto scene is captured at high altitude (20Km) with larger GSD and potential of having several materials in one pixel, which may lead to classification error and hence artefacts in material allocation. Table 6.3 shows that CHIMES TIAEM reports lower error NRMSE error with respect to GT in all five ROIs shown in Figure 5.1. Also in ROIs 2–4 TIAEM has lowest ℓ_1 -norm error.

Despite observing errors in material allocation, the Eigenvectors' plots show relative similarity in first two Eigenvectors with a mean NRMSE of 28% and 48.3%. The third eigenvector has a mean NRMSE of 163.7% despite significant eigenvalue which is also reflected in higher NRMSE of ROIs. We selected the first four Eigenvectors because of the relatively higher magnitude of the corresponding eigenvalues in the first three which is also reported in [86]. The fourth one is selected despite lower eigenvalues to present the deviation in simulation data to GT, which typically appear due to noise [101]. The first Eigenvector plots of covariance matrix TIAEM reports higher NRMSE error that is, 29.7% compared to 26% of BOAEM and 28.3% of CameoSim, respectively. The same trend of error magnitude repeats in second to fourth eigenvector plot, as well.

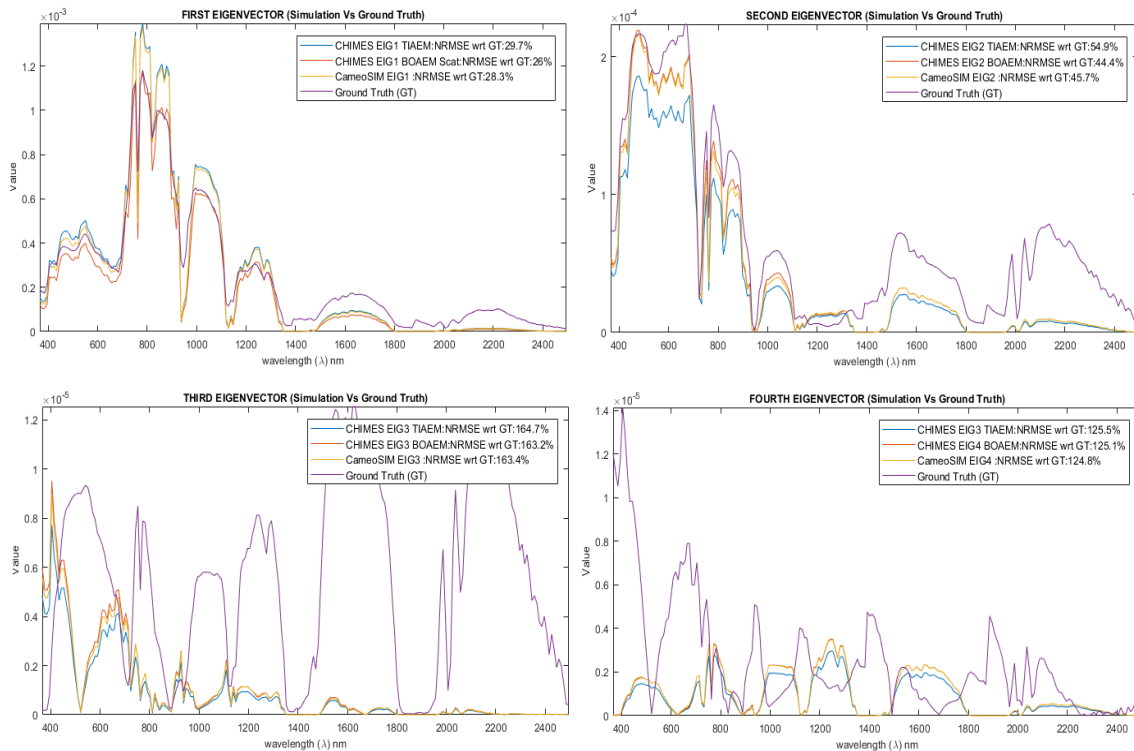


Figure 6.15: Eigenvectors of covariance of whole simulated and GT scenes. NRMSE of simulated scene Eigenvectors with respect to GT's

6.4 Shadow Detection Results

6.4.1 Selene Scene

Firstly, the algorithm is executed on the Selene SCI H23 scene for both global and local search sub-phases of the regression learning. This scene has several white and black calibration panels planted, providing us with a GT for comparison of estimated global L_w and L_b .

6.4.1.1 Result of Regression Learning on Whole Image (Global Search)

The comparative result for global search sub-phase is shown in Figure 4.3. The Normalized Root Mean Square Error (NRMSE) for L_w is 20.62%, which shows that the algorithm can reconstruct the white panel with substantial accuracy. The deviation in the blue region shows under-estimation of scattering/sky radiance, while lower magnitude in NIR is due to over-estimation of the adjacency effect, which is primarily caused by the abundance of vegetation in the background scene. The NRMSE for L_b is 76.66%, which is higher than L_w , it shows that the adjacency effect is over-estimated, hence the estimated black panel looks similar to the vegetating signature. These errors were incurred due to QUAC reflectance, which is the reference for calculating the L_w and L_b in our algorithm.

6.4.1.2 Results of Kernel-Based Regression Learning (Local Search)

Figure 4.4a shows the GT shadow map generated from ray-tracer on DSM. Figure 4.4b, c show results of local search sub-phase. In the case of former, the algorithm was run with a coarse kernel (10×10) and the latter with a fine (3×3) kernel. Better separation of shadowed and non-shadowed regions is visible with 3×3 kernel compared to its coarse counterpart.

In case of the fine kernel, shadows due to bumps on the terrain are also captured which seems to be missing from the coarser one.

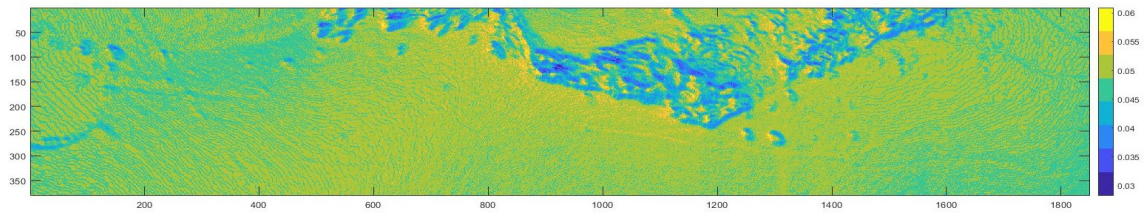
6.4.1.3 Results of Classification

- *Intermediate and Binary maps*

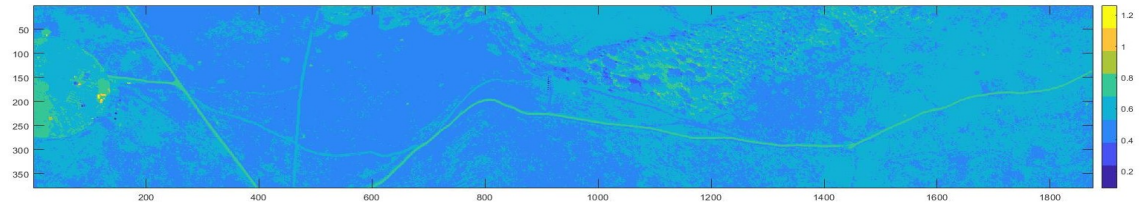
Algorithms adopted for comparative analysis create an intermediate map and apply manual threshold as suggested by respective authors. We propose a cut-off value

of 1 due to the ratio between shadow and non-shadow radiance. Intermediate maps of all algorithms for the Selene scene are shown in Figure 6.16. After applying respective threshold values, we get binary classification maps for the Selene scene as shown in Figure 6.17. The Selene scene has shadowed regions near trees and concrete fields. Concrete field has some buildings that are casting shadows on it. The grass field has a few bumps and has some stand-alone tree distribution along the track.

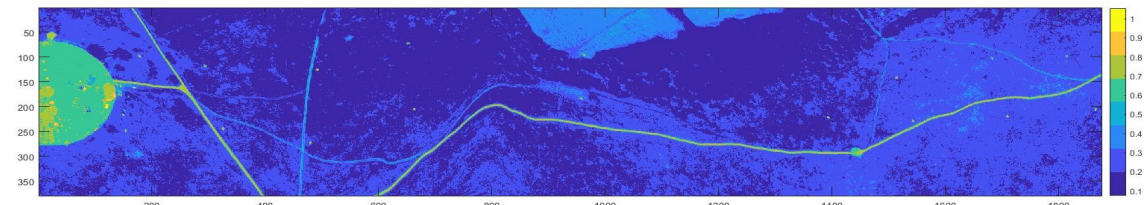
Binary map for the Selene scene GT is constructed by applying a threshold on the DSM contour data. Similarly, the binary map of all competitive algorithms is also constructed. As, the area of the non-shadowed region is significantly larger, covering 92.6377% of the scene, compared to the shadowed region, covering 7.3623% as estimated from the DSM binary map, we divide by the cardinality in the table to minimize bias in overall percentage accuracy.



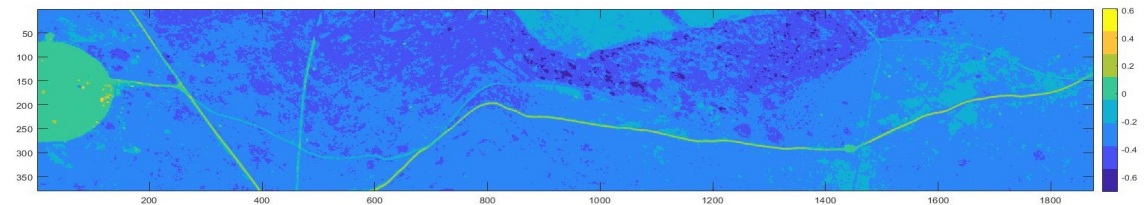
(a) Ground truth (mean of all bands in shadow map created by ray-tracing on DSM geometry)



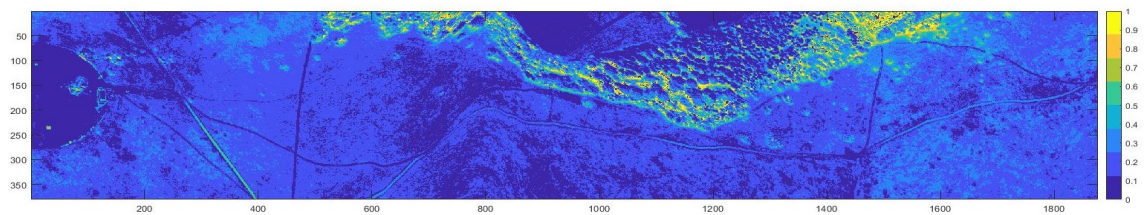
(b) Gevers' Algorithm [84]



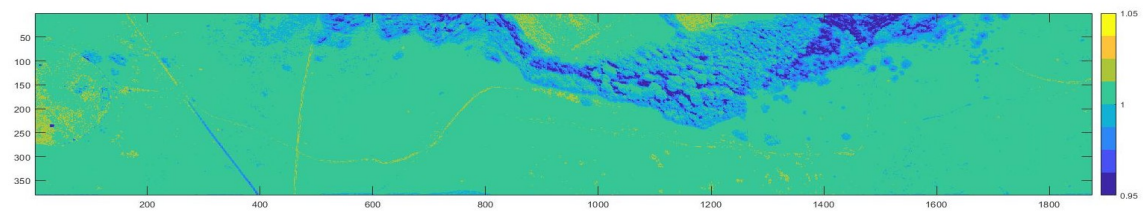
(c) RGBN Algorithm [82]



(d) Beril's Algorithm [79]

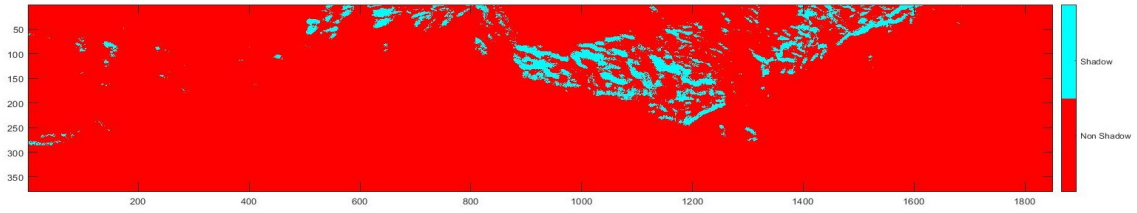


(e) LULC Algorithm [81]

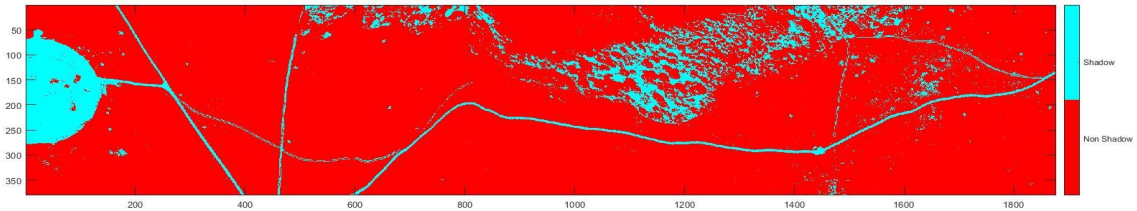


(f) Proposed Algorithm

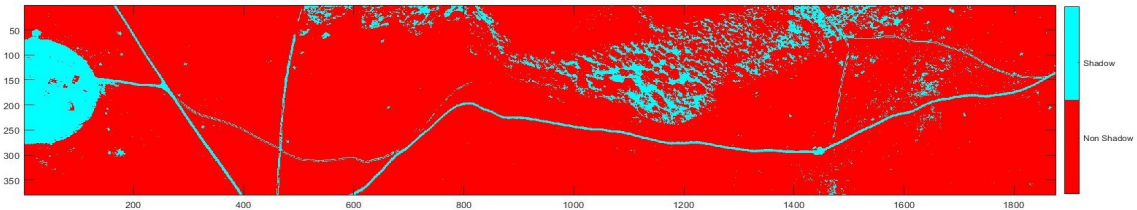
Figure 6.16: SELENE: Intermediate maps extracted by shadow detection methods before thresholding for a binary classification map.



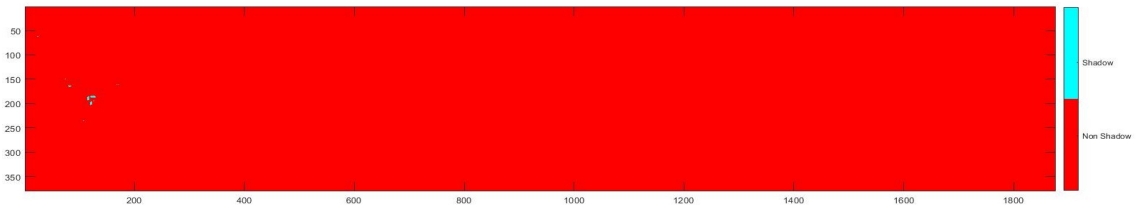
(a) Ground Truth: (Thresholding the Ground Truth map of Figure 6.16 (a) creates a binary map.)



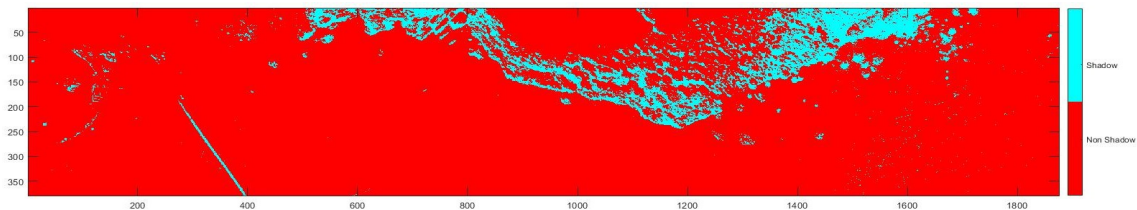
(b) Gevers: It detects roads and concrete field as shadows because they have lower reflectance.



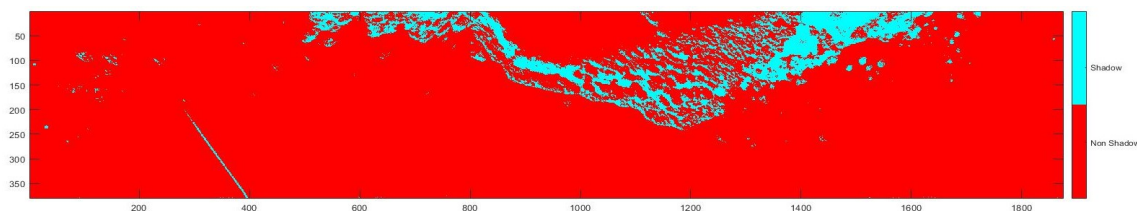
(c) RGBN: Similar to Gevers, it detects roads and concrete field as shadows.



(d) Beril: It only detects some building shadows on the concrete field.



(e) LULC: It detects shadows on both vegetation and concrete field regions, as expected. A portion of road is also detected as shadow.



(f) Proposed: Similar to LULC, it also detects shadows on both vegetation and concrete field regions. Similarly, a portion of road is also detected as shadow, but it is slightly thinner.

Figure 6.17: SELENE: Binary classification map separating shadow and non-shadow regions.

6.4.2 Modesto Scene

The Modesto scene does not have DSM data, therefore, ground truth for this scene is not available. Moreover, it has cloud shadows which may not be addressed by DSM. So, the evaluation, in this case, is rather qualitative, based on visual perception. Intermediate maps of all algorithms of interest for the Modesto scene are shown in Figure 6.18. Their binary counterparts are in Figure 6.19. The proposed algorithm appears to classify the scene into three regions: (i) with clouds, (ii) with shadows, and (iii) the lit ground region. After the threshold is applied to the intermediate map, the scene is classified into shadowed and non-shadowed regions, as shown in Figure 6.19.

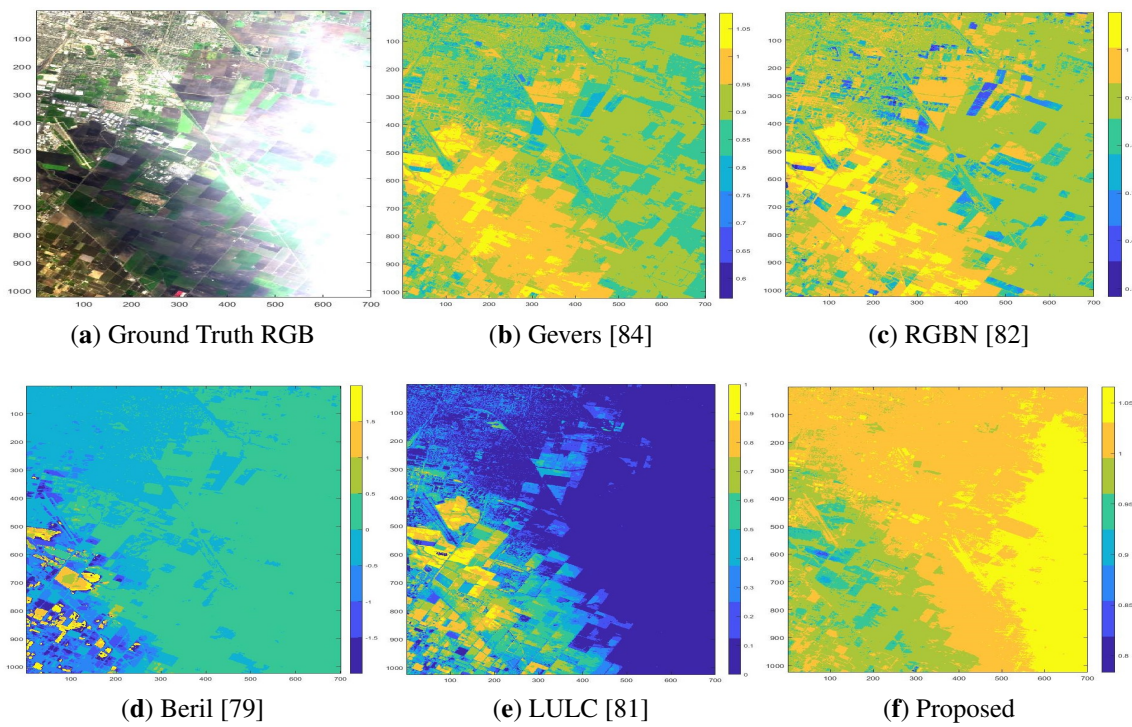


Figure 6.18: MODESTO: Intermediate maps extracted by shadow detection methods before thresholding for a binary classification map.

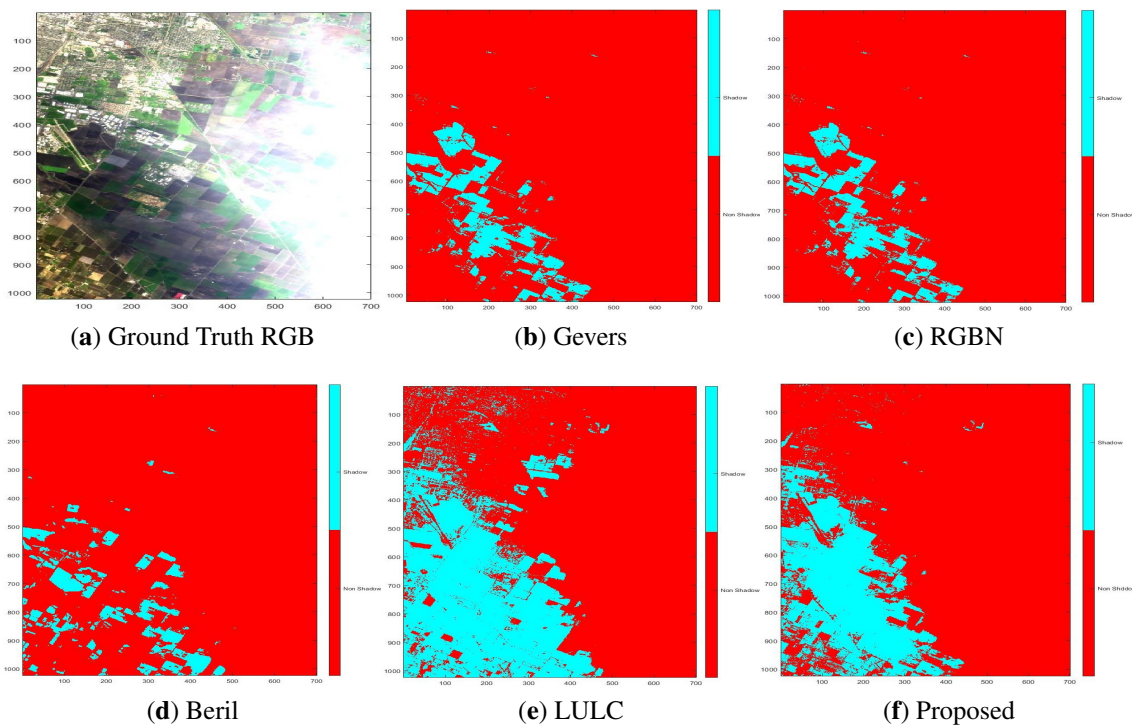


Figure 6.19: MODESTO: Binary classification map (shadow and non-shadow regions.)

Chapter 7

Discussion and Drawbacks

7.1 Scene Simulation

7.1.1 Variable Atmospheric Condition

In a clear-sky atmosphere, MODTRAN's predefined "no-aerosol" setting is used. Clearsky simulation results show that both models of CHIMES produce similar results compared to CameoSim. The ℓ_1 -norm error range for the whole scene simulated with TIAEM and BOAEM the error ranges are between 0–12.79 and 0–12.70 with the mean error of 4.28 and 4.23, respectively. In the case of CameoSim simulation the range of ℓ_1 -norm error is between 0 and 14.71 with a mean error of 4.90. In terms of rendering the coupling of ray tracer with phase-function is relatively insignificant because of lower scattering coefficients in this atmosphere. On the other hand coupling of the ray-tracer direct emitter with target reflectance, BRDF is more dominant which manifests the fact that all models are working similarly when direct light is dominant and atmosphere has lower scattering efficiency. In clear-sky simulations, flat terrain was used. Results of overcast condition are covered in subsequent sections, for flat and DEM terrains.

7.1.2 Flat Terrain

When simulations were performed with flat terrain, CHIMES TIAEM simulated image has ℓ_1 -norm error for the whole scene in the range of 0–3.99 with the mean error of 1.33, compared to CHIMES BOAEMs 0–3.16 with the mean error of 1.05. CameoSim's ℓ_1 -norm error range lies between 0–3.74 with a mean error of 1.25. Therefore CHIMES BOAEM model is the best performer, while TIAEM is the worst in terms of ℓ_1 -norm error, which might be a consequence of higher textural variation during volumetric scattering. On the other hand, if we compare the eigenvectors of the covariance matrix of the simu-

lated scenes, TIAEM's first vectors have minimum 18% NRMSE compared to BOAEMs 79.4% and CameoSims 73.4%, with respect to the ground truth. Better estimation of covariance statistics of a scene signifies better estimation of detector performance through a simulated scene. These target detectors include Adaptive Cosine Estimator (ACE) and Match Filter and so forth, that depends upon HSI image's band to the band covariance matrix. If a simulator generates good statistics of the scene for variable conditions then target vulnerability can be better assessed.

7.1.3 DEM–Based Realistic Terrains

When simulations were performed with DEM–based realistic terrain, all three models show an increase in the mean ℓ_1 -norm error for the whole image. CHIMES BOAEM still possesses the least mean error of 1.09 with a range of 0–3.28. The TIAEM model's ℓ_1 -norm error lies within 0–4.13 with the mean error of 1.38. CameoSim, on the other hand, shows an unexpected many-fold increase in ℓ_1 -norm error compared to both models of CHIMES. As CameoSim is a closed software system, it is not possible to verify the cause of increase error in CameoSim. However, as mentioned in the text the sky-view factor, in this case, becomes less than 1.0, causing non-zero contribution of background reflected radiance.

7.1.4 Variable Altitudes

Another air-borne dataset was used to verify the difference in simulation at varying altitude. The Modesto dataset was acquired by the platform at 20 Km altitude. Due to the increase in altitude, the R value of TIAEM model increased and the upwelled scattered radiance did not show a textured imprint as shown in Figure 3.23 (b). The TIAEM model has the least ℓ_1 -norm error for the whole image with a mean error of 0.81 and the error range of 0–2.42. The BOAEM mode has the mean ℓ_1 -norm error of 0.86 in the range of 0–2.59. The CameoSim results have the highest error with the mean of 0.90 in the range of 0–2.71. A similar trend is a scene in the NRMSE error of different ROIs across the scene. Lower error due to higher altitude and larger area of effective adjacency shows that localised adjacency model is not an realistic fit for the given scene, for low altitudes.

7.1.5 Drawbacks in Scene Simulation

CHIMES simulator has a number of limitations some of them are also defined in the scope of this research.

- Thermal radiance and scattering are not supported.
- Orientation of the sensor can only be the nadir.
- Only sun and moon are supported emitter types.
- Atmospheric generation is not part of the simulator and it is generated through scripts.

7.2 Shadow Detection

The Selene scene GT shadow map enables us to quantify the results of comparative algorithms in context. The proposed algorithm is shown to have 40.46% true positive, i.e., correct detection of a shadow region compared to other counterparts. In this case, the LULC algorithm is marginally inferior to our algorithm, yielding 39.52%. Beril, RGBN, and Gevers reach 0%, 27.14%, and 28.90%, respectively. In terms of false positive, i.e., detecting non-shadow as a shadow, our algorithm continues to perform better than other algorithms, reaching 64% compared to 72.64% of LULC, 112.14% of Gevers, and 108.94% of RGBN. Although Beril shows a lower value of 0.24%, it should be noted that it is the most biased performer, detecting 98% of the region as non-shadowed. Ignoring the Beril algorithm due to bias, our algorithm tops the true-negative detection as well, i.e., detecting non-shadowed region correctly. Our result is 93.49% compared to 92.8%, 89.9%, and 89.7% of LULC, RGBN, and Gevers algorithms, respectively. Finally, our algorithm is also the best performer in a false-negative, i.e., detecting non-shadow as a shadow, yielding only 4.62% compared 4.69%, 5.68%, and 5.54% of LULC, RGBN, and Gevers algorithms, respectively. We conclude that LULC is marginally inferior to our algorithm while others are completely outperformed.

For the Modesto scene, a qualitative evaluation was performed, and it looks like our algorithm performed reasonably well in this dataset.

7.2.1 Drawbacks in Shadow Detection

The pre-requisite of shadow detection algorithm to have QUAC based AC reflectance image incurs errors due to QUAC algorithm. For example, QUAC does not compensate contributions due to adjacency effect. When QUAC reflectance and radiance are used to compute TGRR and path radiance, then this error will be accumulated in them and cause discrepancies. It is also observed that the reconstructed radiance also possess noise in some pixels, particularly in the VNIR region. Although it is overcome by non-linear filtering, however a more physics-based method for reducing this noise should be found. Moreover, the final shadow-map is dependent on a user-defined threshold, which may vary algorithms performance from user to user.

Evaluation of shadow detection on a large outdoor scene is always a challenge because typically the Ground Truth is unknown. As an alternate, we employed the DEM data and ran a ray-tracing algorithm to cast the shadow. This shadow-map is used as the GT which may not be a good representation of realistic shadows in the scene. This is why the comparative results of shadow detection are included in the Appendix instead.

Chapter 8

Conclusion And Future Work

In this project, a great deal of effort is placed in establishing an HSI scene simulator code-base. In this chapter, we will revisit the problem statement where well-defined goals were set to evaluate the degree of achievement. Performance of the simulator is also discussed with regards to important metrics such as variable atmospheric conditions, adjacency effect models, terrain types and sensor altitudes.

8.1 Achievements

The primary objective of this research was to develop a software which will enable Cranfield University access to comprehensive code-base and detailed understanding of the significant factors involved in hyperspectral scene simulation. The software development activities of this work were solely performed by the author of this report. Having this immense resource constrain, several open-source libraries were reused to accelerate the development process such that more time and focus is given to the research content and development of new features that are lacking in the COTS simulators.

We are successful in the development of a new scene simulation software package whose radiance results are comparable with CameoSim and even perform better when DEM map based terrain is generated. Moreover, CHIMES generates rendering based image where volumetric scattering is also employed.

One of the most challenging factors in generating simulation images comparable to ground truth is the ability to reconstruct realistic atmospheric data. A more difficult part in the reconstruction is to capture the scene's textural and altitude dependent heterogeneity. Existing COTS simulators discussed in section 2.2.1 leave atmospheric data generation to the users. This is an immense challenge to find parameters that match the ground truth scene.

The need to automate such an atmospheric parameter search was realized during the inception phase when we used CameoSim to generate atmospheric data. Therefore such automation is a novel and integrated part of CHIMES simulator.

DISORT is a well-established algorithm for generating scattering data along with background surface albedo based adjacency effect incorporation. It is used by MODTRAN, however, a serious limitation is in the support for a single albedo reflectance-based adjacency effect. A typical use-case of DISORT in MODTRAN is to apply the average reflectance of a scene as the background albedo. However, this approach fails to capture the localisation factor of adjacency effect, which is reported in [22].

An emphasis is given to the localised adjacency effect model in CHIMES. Due to localisation, our adjacency model is integrated into the renderer, which is also unprecedented in comparison with COTS. It is particularly observed that the covariance data generated with localised adjacency model is much closer to GT compared to one reflectance-based MODTRAN/DISORT's calculations.

During the course of the data analysis in the project, we realised that a larger error is in the regions of images where shadows are persistent. The aim, therefore, was to be able to compensate those shadows, however, we could not proceed further due to time limitations.

Shadow detection is a tentative phase of shadow compensation. After research, we found that existing shadow algorithms lack radiative transfer background and so we developed a shadow detection algorithm based on RT theory.

8.2 CHIMES Simulator

CHIMES is a rendering-based HSI scene simulator with unique features of renderer integrated localised adjacency effect model and automated atmospheric search. It also includes volumetric scattering, rugged terrain generation based on DEM data and shadow detection support.

We implemented a new end-to-end Hyperspectral Image generation system, which generates HSI image in the visible to SWIR region of the electromagnetic spectrum. The image simulated in this work was taken by Hypspec dual VNIR1600 and SWIR384 cameras. Our simulator creates a skydome from MODTRAN radiance and employs path tracing based light transport for calculating the at-sensor radiance. It also incorporates Digital Elevation Models to generate the relatively rugged surface. We also presented a method to search for atmospheric parameters, given a known reflectance material in the scene.

We proposed two adjacency effect models one which uses an average background reflectance for the whole scene (BOAEM), we also proposed localized background spectra retrieved from neighbouring texture pixels (TIAEM). Both of our models employ volumetric backscattering builtin the renderer. We compared results of both these models with CameoSim and Ground Truth.

8.2.1 Methodologies

A radiative transfer model-based approach for scene simulation is the most popular and adopted in the research community. There are several components of radiations that accumulate to construct an image. A comprehensive Big equation by Rochester Institute of Technology attempts to capture all these components as additive terms of the equation in an image chain approach. We have modified their RT equation to incorporate our localized adjacency model which is an integral part of the renderer. Moreover, volumetric scattering is performed with an integrated phase function with the renderer for diffuse radiance computation. The rendering process is, therefore, volumetric path tracing, spectral rendering is captured by providing all relevant data such as reflectance, BRDF, MODTRAN radiance etc. as vectors. Therefore the renderer is completely vectorized and accelerated according to computer architecture for improved time performance. Other aspects of CHIMES performance are described in the subsequent section.

8.2.2 Performance of Simulator

Imagery generated by CHIMES using BOAEM model shows close agreement with CameoSim. The CHIMES TIAEM model shows deviation from both BOAEM and CameoSim results and yields a higher error in flat scene geometry, however, it consistently performed better in eigenanalysis of the whole imagery, particularly in first eigenvectors of both flat and DEM terrains. Moreover, analysis of target material in Appendix A shows that the shape of radiance generated by CHIMES TIAEM matches closely in VNIR region. It is noted during the analysis that CHIMES-BOAEM came out to have the least error as shown in Table 6.1, it even showed a slump in error when rugged terrain is used, as shown in Table 6.2. This DEM data represent the real height-map of the scene when it was captured.

CameoSim generated imagery in flat terrain stands second in terms of least error with respect to ground truth, as shown in Table 6.1. However, when DEM is introduced in terrain, the error leapt up by several times as shown in Figure 6.7 and Table 6.2. Selene

scene has some regions consisting tall tree, however, it is essentially a flat scene otherwise. An increase in error to this extent was therefore not expected.

In the case of Modesto scene TIAEM turns out to be the best model with minimum ℓ_1 -norm error, it is potentially because the R is larger and the texture imprint in the upwelled radiance is not significant.

8.2.3 Future Work

CHIMES simulator can be improved in several ways, some of the features should be added and a more comprehensive testing should be performed.

- Support for arbitrary sensor orientation should be implemented.
- Thermal radiance and scattering model should included.
- Only Lambertian and Phong BRDF models well tested. Cook and Torrence should be tested and new models should be added.
- A more functional GUI should be implemented. Loading of parameters from XML file should be through GUI based control for ease of use.
- BRDF models should be tested against MERL benchmark datasets.
- Scene simulation performance should be tested by standardizing RAMI [102] datasets.
- Simulator should be tested against more existing simulators for better intercomparison and validation.

8.3 Shadow Detection

Presence of white and black calibration panel in the Selene scene enables us to evaluate the results of the regression learning phase. The recovery of TGRR approaches 79% accuracy. Conversely, path radiance recovery proved to be more error-prone due to smaller magnitude and contribution in at-sensor radiance. It is observed that radiance in the SWIR region for TGRR is notably higher than ground truth. Presence of abundant vegetation in the scene shows a potential adjacency effect, which is causing the raised magnitude because pixels from many points across the scene are employed for the recovery. The

goodness of fit of MODTRAN based TGRR with respect to regression's shows higher errors during multiple runs of MODTRAN. The best fit has NRMSE error of 25% which is comparable with the GT white panel, whereas the MODTRAN white panel has only 9% of error compared to GT white panel.

8.3.1 Future Work

The following potential improvements in our existing work are recommended.

- Establishing a controlled environment for creating realistic GT measurements of the shadows to improve the performance analysis of the algorithm.
- Dependence of the shadow-map creation on a user-defined threshold must be transformed into a more adaptive approach to have consistency in the shadow detection in varying user experience.
- Due to errors in the input to algorithm due to QUAC AC processing the estimated entities could have discrepancies. A post-processing method to QUAC, particularly for adjacency effect compensation, could increase the detection performance.
- The shadow detection algorithm in localized mode requires long computations. The algorithm should be made more efficient from timing and memory requirements.

8.3.2 Project Management

The total research grant and the time allocated for this work was 3.5 years. The first-year of this project was served in the literature review, understanding of scene simulation by CameoSim, feature and parameters of MODTRAN etc. Three months were spent on controlled full and sub-pixel planting of a target in CameoSim environment as a project deliverable to the sponsor.

One year and nine months were spent on the development of complete Simulator with localised adjacency, rendering, volumetric scattering, automated atmospheric search features. Furthermore, two months were spent on the shadow detection work which was done in collaboration with a colleague.

One month of effort at the beginning of the PhD was also spent in a project funded by P&G for Washing powder ingredients analysis by multi-spectral imaging, together with relevant literature studies.

References

- [1] C. J. Willers, *Electro-Optical System Analysis and Design: A Radiometry Perspective*. SPIE, mar 2013.
- [2] W. Matusik, H. Pfister, M. Brand, and L. McMillan, “A data-driven reflectance model,” in *ACM SIGGRAPH 2003 Papers*, SIGGRAPH '03, (New York, NY, USA), p. 759–769, Association for Computing Machinery, 2003.
- [3] B. M. Matusik W., Pfister H. and M. L., “Mitsubishi electric research laboratory brdf database, version 2,” 2012.
- [4] C. J. Willers, M. S. Willers, and F. Lapierre, “Signature modelling and radiometric rendering equations in infrared scene simulation systems,” in *Technologies for Optical Countermeasures VIII* (D. H. Titterton and M. A. Richardson, eds.), vol. 8187, pp. 173–188, International Society for Optics and Photonics, SPIE, 2011.
- [5] J. R. Schott, “Remote Sensing, The Image Chain Approach. 1997,” *Oxford University Press*, pp. 80–92.
- [6] H. Zhao, C. Jiang, G. Jia, and D. Tao, “Simulation of hyperspectral radiance images with quantification of adjacency effects over rugged scenes,” *Measurement Science and Technology*, vol. 24, p. 125405, nov 2013.
- [7] C. Jiang, Y. Bao, K. Xing, L. Gao, and F. Li, “Estimation of the adjacency effect with multiple scattering in the simulated signals observed over rugged areas,” in *Remote Sensing of Clouds and the Atmosphere XXI* (A. Comerón, E. I. Kassianov, and K. Schäfer, eds.), vol. 10001, pp. 111 – 117, International Society for Optics and Photonics, SPIE, 2016.
- [8] P. H. Christensen and W. Jarosz, “The path to path-traced movies,” *Found. Trends. Comput. Graph. Vis.*, vol. 10, p. 103–175, Oct. 2016.
- [9] G. H. Albert Kirk, Colin Stroud, “Cameosim user guide,” 2017.

- [10] J. Piper, “A new dataset for analysis of hyperspectral target detection performance,” in *HSI 2014, Hyperspectral Imaging and Applications Conference, October 2014, Coventry UK.*, oct 2014.
- [11] K. Stamnes, S.-C. Tsay, W. Wiscombe, and K. Jayaweera, “Numerically stable algorithm for discrete-ordinate-method radiative transfer in multiple scattering and emitting layered media,” *Appl. Opt.*, vol. 27, pp. 2502–2509, jun 1988.
- [12] . M. T. Eismann, *Hyperspectral Remote Sensing*. SPIE Press, SPIE Press, 2012.
- [13] S. J. R. B. Goetz AF, Vane G, “Imaging spectrometry for earth remote sensing., year = 1985, journal = Science, doi = doi:10.1126/science.228.4704.1147, pages = 1147–1153,”
- [14] M. J. Barnsley, J. J. Settle, M. A. Cutter, D. R. Lobb, and F. Teston, “The proba/chris mission: a low-cost smallsat for hyperspectral multiangle observations of the earth surface and atmosphere,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 42, no. 7, pp. 1512–1520, 2004.
- [15] E. S. Agency, “eoportal,” 2020.
- [16] L. Guanter, K. Segl, and H. Kaufmann, “Simulation of Optical Remote-Sensing Scenes With Application to the EnMAP Hyperspectral Mission,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, pp. 2340–2351, jul 2009.
- [17] Cubert-GMBH, *Cubert S185 brochure*. 2020.
- [18] U. A. Zahidi, P. W. T. Yuen, J. Piper, and P. S. Godfree, “An end-to-end hyperspectral scene simulator with alternate adjacency effect models and its comparison with cameosim,” *Remote Sensing*, vol. 12, no. 1, 2019.
- [19] J. P. Kerekes and J. E. Baum, “Full-spectrum spectral imaging system analytical model,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, pp. 571–580, mar 2005.
- [20] J. R. Schott, *Remote sensing: the image chain approach*. Oxford University Press on Demand, 2007.
- [21] R. Richter, “ATCOR version 4, Software help documents,” *Web link*, 2016.

- [22] S. Sterckx, E. Knaeps, and K. Ruddick, "Detection and correction of adjacency effects in hyperspectral airborne data of coastal and inland waters: the use of the near infrared similarity spectrum," *International Journal of Remote Sensing*, vol. 32, no. 21, pp. 6479–6505, 2011.
- [23] F. Vagni, "Survey of hyperspectral and multispectral imaging technologies. technical report," 2007.
- [24] U. Navy, "Naval earthmap observer," 2010.
- [25] J. P. Kerkes, "Introduction to FASSP and List of Related Publications," *Web link*, 2005.
- [26] J. R. Schott, S. D. Brown, R. V. Raqueño, H. N. Gross, and G. Robinson, "An Advanced Synthetic Image Generation Model and its Application to Multi/Hyperspectral Algorithm Development," *Canadian Journal of Remote Sensing*, vol. 25, no. 2, pp. 99–111, 1999.
- [27] E. J. Ientilucci and S. D. Brown, "Advances in wide-area hyperspectral image simulation," in *Targets and Backgrounds IX Characterization and Representation* (W. R. Watkins, D. Clement, and W. R. Reynolds, eds.), vol. 5075, pp. 110–121, International Society for Optics and Photonics, SPIE, 2003.
- [28] E. J. Ientilucci, "Synthetic Simulation and Modeling of Image Intensified CCDs (IICCD)," 1996.
- [29] A. A. Goodenough and S. D. Brown, "DIRSIG5: Next-Generation Remote Sensing Data and Image Simulation Framework," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, pp. 4818–4833, nov 2017.
- [30] A. Berk, P. Conforti, R. Kennett, T. Perkins, F. Hawes, and J. van den Bosch, "MODTRAN6: a major upgrade of the MODTRAN radiative transfer code," in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* (M. Velez-Reyes and F. A. Kruse, eds.), vol. 9088, pp. 113–119, International Society for Optics and Photonics, SPIE, 2014.
- [31] A. Brady and S. Kharabash, "Further Studies into Synthetic Image Generation using CameoSim," 2011.

- [32] I. R. Moorhead, M. A. Gilmore, A. W. Houlbrook, D. E. Oxford, D. R. Filbee, C. A. Stroud, G. Hutchings, and A. Kirk, "CAMEO-SIM: a physics-based broadband scene simulation tool for assessment of camouflage, concealment, and deception methodologies," *Optical Engineering*, vol. 40, no. 9, pp. 1896–1905, 2001.
- [33] C. Nelsson, P. Hermansson, T. Winzell, and S. Sjökvist, "Benchmarking and Validation of IR Signature Programs: SensorVision, CAMEO-SIM and RadThermIR," 2005.
- [34] M. S. Willers and C. J. Willers, "Key considerations in infrared simulations of the missile-aircraft engagement," in *Technologies for Optical Countermeasures IX* (D. H. Titterton and M. A. Richardson, eds.), vol. 8543, pp. 180–195, International Society for Optics and Photonics, SPIE, 2012.
- [35] C. J. Willers, M. S. Willers, and A. de Waal, "Aircraft vulnerability analysis by modeling and simulation," in *Technologies for Optical Countermeasures XI; and High-Power Lasers 2014: Technology and Systems* (D. H. Titterton, M. A. Richardson, R. J. Grasso, W. L. Bohn, and H. Ackermann, eds.), vol. 9251, pp. 151–166, International Society for Optics and Photonics, SPIE, 2014.
- [36] R. L. Sundberg, S. Richtsmeier, and R. Haren, "Full optical spectrum hyperspectral scene simulation," in *Proceedings. 2005 IEEE International Geoscience and Remote Sensing Symposium, 2005. IGARSS '05.*, vol. 5, pp. 3235–3238, jul 2005.
- [37] R. Sundberg, S. Richtsmeier, A. Berk, S. M. Adler-Golden, M. J. Fox, and R. Haren, "Thermal infrared scene simulation for plume detection algorithm evaluation," in *SPIE Defense + Commercial Sensing*, 2004.
- [38] W. Pereira, S. Richtsmeier, S. Carr, S. Kharabash, and A. Brady, "A comparison of MCScene and CameoSim simulations of a real scene," in *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–4, jun 2014.
- [39] F. T. Hawes, A. Berk, and S. C. Richtsmeier, "Development and validation of P-MODTRAN7 and P-MCScene, 1D and 3D polarimetric radiative transfer models," in *Polarization: Measurement, Analysis, and Remote Sensing XII* (D. B. Chenault and D. H. Goldstein, eds.), vol. 9853, pp. 215–221, International Society for Optics and Photonics, SPIE, 2016.

- [40] S. A. Cota, J. T. Bell, R. H. Boucher, T. E. Dutton, C. J. Florio, G. A. Franz, T. J. Grycewicz, L. S. Kalman, R. A. Keller, T. S. Lomheim, D. Paulson, and T. S. Wilkinson, "PICASSO: an end-to-end image simulation tool for space and airborne imaging systems," *Journal of Applied Remote Sensing*, vol. 4, no. 1, pp. 1–37, 2010.
- [41] S. A. Cota, T. S. Lomheim, C. J. Florio, J. M. Harbold, B. M. Muto, R. B. Schoolar, D. T. Wintz, and R. A. Keller, "PICASSO: an end-to-end image simulation tool for space and airborne imaging systems II. Extension to the thermal infrared: equations and methods," in *Imaging Spectrometry XVI* (S. S. Shen and P. E. Lewis, eds.), vol. 8158, pp. 160–184, International Society for Optics and Photonics, SPIE, 2011.
- [42] S. Chandrasekhar, *Radiative Transfer*. Dover books on advanced mathematics, Dover Publications, 1960.
- [43] M. Born and E. Wolf, *Principles of Optics Electromagnetic Theory of Propagation, Interference and Diffraction of Light*. 1980.
- [44] S. Chandrasekhar, *Optics of the Atmosphere*. Wiley series in pure and applied mathematics, John Wiley, 1976.
- [45] R. S. John R. Howell, M. Pinar Mengüç, *Thermal Radiation Heat Transfer*. CRC Press, CRC Press, 2015.
- [46] H. Hulst, *Light Scattering by Small Particles*. Structure of matter series, Wiley, 1957.
- [47] L. P. Corson, Dale R., *Electromagnetic Fields and Waves*. CRC Press, W.H.Freeman Co. Ltd, 1962.
- [48] M. Pharr, W. Jakob, and G. Humphreys, *Physically based rendering: From theory to implementation*. Morgan Kaufmann, 2016.
- [49] E. Veach, *Robust Monte Carlo methods for light transport simulation*, vol. 1610. Stanford University PhD thesis, 1997.
- [50] B. Dobbs, "The Incorporation of Atmospheric Variability into DIRSIG," *RIT Master's Thesis*, 2006.

- [51] J. Otterman and R. S. Fraser, "Adjacency effects on imaging by surface reflection and atmospheric scattering: cross radiance to zenith," *Appl. Opt.*, vol. 18, pp. 2852–2860, Aug 1979.
- [52] J. Dave, "Effect of atmospheric conditions on remote sensing of vegetation parameters," *Remote Sensing of Environment*, vol. 10, no. 2, pp. 87 – 99, 1980.
- [53] D. Tanré, P. Y. Deschamps, P. Duhaut, and M. Herman, "Adjacency effect produced by the atmospheric scattering in thematic mapper data," *Journal of Geophysical Research: Atmospheres*, vol. 92, no. D10, pp. 12000–12006, 1987.
- [54] D. Tanre, M. Herman, and P. Y. Deschamps, "Influence of the background contribution upon space measurements of ground reflectance," *Appl. Opt.*, vol. 20, pp. 3676–3684, oct 1981.
- [55] Y. J. Kaufman, "The atmospheric effect on the separability of field classes measured from satellites," *Remote Sensing of Environment*, vol. 18, no. 1, pp. 21–34, 1985.
- [56] P. M. M. Groetsch, P. Gege, S. G. H. Simis, M. A. Eleveld, and S. W. M. Peters, "Variability of adjacency effects in sky reflectance measurements," *Opt. Lett.*, vol. 42, pp. 3359–3362, Sep 2017.
- [57] G. P. A. A. Berk and P. K. Acharya, "MODTRAN 5.3.2 USER'S MANUAL," *SPECTRAL SCIENCES, INC., Burlington MA*, 2013.
- [58] A. A. Goodenough, "Discussion of requirements for external atmospheric data and models," *Web link*, 2016.
- [59] R. Richter, "Atmospheric / Topographic Correction for Airborne Imagery (ATCOR4 User Guide)," *Web link*, 2016.
- [60] Ma Jianwen, Li Xiaowen, Chen Xue, and Feng Chun, "Target adjacency effect estimation using ground spectrum measurement and Landsat-5 Satellite data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, pp. 729–735, mar 2006.
- [61] W. D. Rudolf Richter Martin Bachmann and A. Müller, "Influence of the Adjacency Effect on Ground Reflectance Measurements," *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 4, pp. 1–12, 2006.

- [62] K. Perlin and E. M. Hoffert, "Hypertexture," *SIGGRAPH Comput. Graph.*, vol. 23, p. 253–262, July 1989.
- [63] B. G. S. M. A.-G. Lawrence S. Bernstein Xuemin Jin, "Quick atmospheric correction code: algorithm description and recent upgrades," *Optical Engineering*, vol. 51, no. 11, pp. 1–12, 2012.
- [64] D. Schläpfer, "MODO: An interface to MODTRAN4 for the simulation of imaging spectrometry at-sensor signals," 2001.
- [65] J. M. Fontenla, J. Harder, W. Livingston, M. Snow, and T. Woods, "High-resolution solar spectral irradiance from extreme ultraviolet to far infrared," *Journal of Geophysical Research: Atmospheres*, vol. 116, no. D20, 2011.
- [66] J. R. Schott, *Fundamentals of Polarimetric Remote Sensing*. SPIE Press: Bellingham Washington USA, mar 2009.
- [67] A. Kokhanovsky, *Optics of Light Scattering Media: Problems and Solutions*. Springer London Ltd; 2nd edition, 2001.
- [68] A. Kokhanovsky, "Optical properties of terrestrial clouds," *Earth-Science Reviews*, vol. 64, no. 3, pp. 189–241, 2004.
- [69] T. Arvidson, J. Gasch, and S. N. Goward, "Landsat 7's long-term acquisition plan — an innovative approach to building a global imagery archive," *Remote Sensing of Environment*, vol. 78, no. 1, pp. 13–26, 2001.
- [70] R. R. Irish, "Landsat 7 automatic cloud cover assessment," vol. 4049, 2000.
- [71] Z. Zhu and C. E. Woodcock, "Object-based cloud and cloud shadow detection in Landsat imagery," *Remote Sensing of Environment*, vol. 118, pp. 83–94, 2012.
- [72] H. Zhang, K. Sun, and L. Wenzhuo, "Object-Oriented Shadow Detection and Removal From Urban High-Resolution Remote Sensing Images," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, pp. 6972–6982, 2014.
- [73] E. Salvador, A. Cavallaro, and T. Ebrahimi, "Shadow identification and classification using invariant color models," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 3, pp. 1545–1548 vol.3, may 2001.

- [74] X. Liu, Z. Hou, Z. Shi, Y. Bo, and J. Cheng, "A shadow identification method using vegetation indices derived from hyperspectral data," *International Journal of Remote Sensing*, vol. 38, pp. 5357–5373, 2017.
- [75] N. N. Imai, A. M. G. Tommaselli, A. Berveglieri, and E. A. S. Moriya, "SHADOW DETECTION IN HYPERSPECTRAL IMAGES ACQUIRED BY UAV," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 371–377, 2019.
- [76] G. Tolt, M. Shimoni, and J. Ahlberg, "A shadow detection method for remote sensing images using VHR hyperspectral and LIDAR data," in *2011 IEEE International Geoscience and Remote Sensing Symposium*, pp. 4423–4426, jul 2011.
- [77] P. Hartzell, C. Glennie, and S. Khan, "Terrestrial Hyperspectral Image Shadow Restoration through Lidar Fusion," *Remote Sensing*, vol. 9, p. 421, 2017.
- [78] A. Sanin, C. Sanderson, and B. C. Lovell, "Shadow detection: A survey and comparative evaluation of recent methods," *Pattern Recognition*, vol. 45, no. 4, pp. 1684 – 1695, 2012.
- [79] B. Sirmacek and C. Unsalan, "Damaged building detection in aerial images using shadow Information," in *2009 4th International Conference on Recent Advances in Space Technologies*, pp. 249–252, jun 2009.
- [80] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, pp. 62–66, jan 1979.
- [81] M. Teke, E. Ba\cseski, A. Ö. Ok, B. Yüksel, and Ç. \cSenaras, "Multi-spectral False Color Shadow Detection," in *Photogrammetric Image Analysis* (U. Stilla, F. Rottensteiner, H. Mayer, B. Jutzi, and M. Butenuth, eds.), (Berlin, Heidelberg), pp. 109–119, Springer Berlin Heidelberg, 2011.
- [82] D. Ö. B. Hazan Dağlayan Sevim Yasemin Yardımcı Çetin, "A novel method to detect shadows on multispectral images," vol. 10004, 2016.
- [83] P. Sarabandi, F. Yamazaki, M. Matsuoka, and A. Kiremidjian, "Shadow detection and radiometric restoration in satellite high resolution images," in *IGARSS 2004. 2004 IEEE International Geoscience and Remote Sensing Symposium*, vol. 6, pp. 3744–3747 vol.6, 2004.

- [84] W. c. b. A. W. M. Smeulders and A. D. Bagdanov, *Color Feature Detection*, ch. 13, pp. 187–220. John Wiley & Sons, Ltd, 2012.
- [85] U. A. Zahidi, A. Chatterjee, and P. W. T. Yuen, “A Radiative Transfer Model-Based Multi-Layered Regression Learning to Estimate Shadow Map in Hyperspectral Images,” *Machine Learning and Knowledge Extraction*, vol. 1, no. 3, pp. 904–927, 2019.
- [86] K. Bloechl, C. D. Angelis, M. Gartley, J. Kerekes, and C. E. Nance, “A comparison of real and simulated airborne multisensor imagery,” in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XX* (M. Velez-Reyes and F. A. Kruse, eds.), vol. 9088, pp. 97 – 112, International Society for Optics and Photonics, SPIE, 2014.
- [87] Z. Cui, J. Kerekes, C. DeAngelis, S. Brown, and C. E. Nance, “A comparison of real and simulated airborne hyperspectral imagery,” in *2014 IEEE Western New York Image and Signal Processing Workshop (WNYISPW)*, pp. 19–22, Nov 2014.
- [88] C. Nelsson, E. Andersson, D. Boerjesson, P. Hermansson, S. Nyberg, S. Sjoekvist, and T. Winzell, “Methods for validation of optical signature models,” in *Targets and Backgrounds XI: Characterization and Representation* (W. R. Watkins, D. Clement, and W. R. Reynolds, eds.), vol. 5811, pp. 212 – 223, International Society for Optics and Photonics, SPIE, 2005.
- [89] K. Fukunaga, *Introduction to statistical pattern recognition (2nd edition)*. Academic Press, 1990.
- [90] C. M. Birkemark, “CAMEVA, a methodology for estimation of target detectability,” *Optical Engineering*, vol. 40, no. 9, pp. 1835 – 1843, 2001.
- [91] L. C. P. Coppo and L. Alparone, “Validation and simulation examples of an end-to-end simulator for optical imaging systems,” in *Sensors, Systems, and Next-Generation Satellites XVII* (R. Meynart, S. P. Neeck, and H. Shimoda, eds.), vol. 8889, pp. 434 – 447, International Society for Optics and Photonics, SPIE, 2013.
- [92] R. A. White, *Validation of Rochester Institute of Technology’s (RIT’s) Digital Image and Remote Sensing Image Generation (DIRSIG) Model - Reflective Region*.

- Rochester Institute of Technology Chester F. Carlson Center for Imaging Science, 1996.
- [93] J. R. S. Scott D. Brown, “Verification and validation studies of the dirsig data simulation model,” 2010.
- [94] C. Lanaras, E. Baltsavias, and K. Schindler, “Estimation of relative sensor characteristics for hyperspectral super-resolution,” in *2016 8th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pp. 1–5, 2016.
- [95] S. O. ”Zhaohui Guo”, “Template matching via l_1 minimization and its application to hyperspectral data,” 2011.
- [96] S. Bakken, M. Orlandic, and T. A. Johansen, “The effect of dimensionality reduction on signature-based target detection for hyperspectral remote sensing,” in *CubeSats and SmallSats for Remote Sensing III* (T. S. Pagano, C. D. Norton, and S. R. Babu, eds.), vol. 11131, pp. 164 – 182, International Society for Optics and Photonics, SPIE, 2019.
- [97] A. Chatterjee and P. W. T. Yuen, “Endmember learning with k-means through scd model in hyperspectral scene reconstructions,” *Journal of Imaging*, vol. 5, no. 11, 2019.
- [98] B. Sirmacek, “Source code for beril’s algorithm. *the mathworks, inc.*,” 2016.
- [99] M. Teke, “Source code for lulc algorithm. *github, inc.*,” 2014.
- [100] I. Ibrahim, P. Yuen, A. Tsitiridis, K. Hong, T. Chen, J. Jackman, D. James, and M. Richardson, “Illumination independent object recognitions in hyperspectral imaging,” in *Optics and Photonics for Counterterrorism and Crime Fighting VI and Optical Materials in Defence Systems Technology VII* (C. Lewis, D. Burgess, R. Zamboni, F. Kajzar, and E. M. Heckman, eds.), vol. 7838, pp. 205 – 216, International Society for Optics and Photonics, SPIE, 2010.
- [101] Y. Chen, D. Wen, L. Jing, and P. Shi, “Shadow information recovery in urban areas from very high resolution satellite imagery,” *International Journal of Remote Sensing*, vol. 28, no. 15, pp. 3249–3254, 2007.

- [102] J.-L. Widlowski, M. Taberner, B. Pinty, V. Bruniquel-Pinel, M. Disney, R. Fernandes, J.-P. Gastellu-Etchegorry, N. Gobron, A. Kuusk, T. Lavergne, S. Leblanc, P. E. Lewis, E. Martin, M. Möttus, P. R. J. North, W. Qin, M. Robustelli, N. Rochdi, R. Ruiloba, C. Soler, R. Thompson, W. Verhoef, M. M. Verstraete, and D. Xie, “Third radiation transfer model intercomparison (rami) exercise: Documenting progress in canopy reflectance models,” *Journal of Geophysical Research: Atmospheres*, vol. 112, no. D9, 2007.
- [103] A. Polukhin, *Boost C++ Application Development Cookbook, Second Edition*. 2017.
- [104] I. Cooperation, “Open-source ray tracing toolkit.*github, inc.*,” 2016.
- [105] C. J. Willers, M. S. Willers, R. A. T. Santos, P. J. van der Merwe, J. J. Calitz, A. de Waal, and A. E. Mudau, “Pyradi: an open-source toolkit for infrared calculation and data processing,” in *Technologies for Optical Countermeasures IX* (D. H. Titterton and M. A. Richardson, eds.), vol. 8543, pp. 125 – 139, International Society for Optics and Photonics, SPIE, 2012.

Appendix A

Material Wise Simulation Results

In Figures 6.4 and 6.8 we presented mean radiance of ROIs given in Figure 5.1. The prime focus in selecting these ROIs is firstly in capturing the background that is, vegetation for most of the targets and secondly concrete on which a number of other target are planted. It is however important to highlight some man made materials such as orange perspex and other patch of concrete, radiance of these targets are shown in this appendix. Radiance of each target is plotted and statistical analysis is presented next to them, along with eigenanalysis. In Figures A.1–A.4 results of radiance with a flat terrain of Selene scene are presented.

Figures A.9–A.11 show material wise ROI simulated radiance plots of Modesto scene with flat terrain. It includes farm soil, mixture of soil and grass and an ROI covering the urban region of Modesto town.

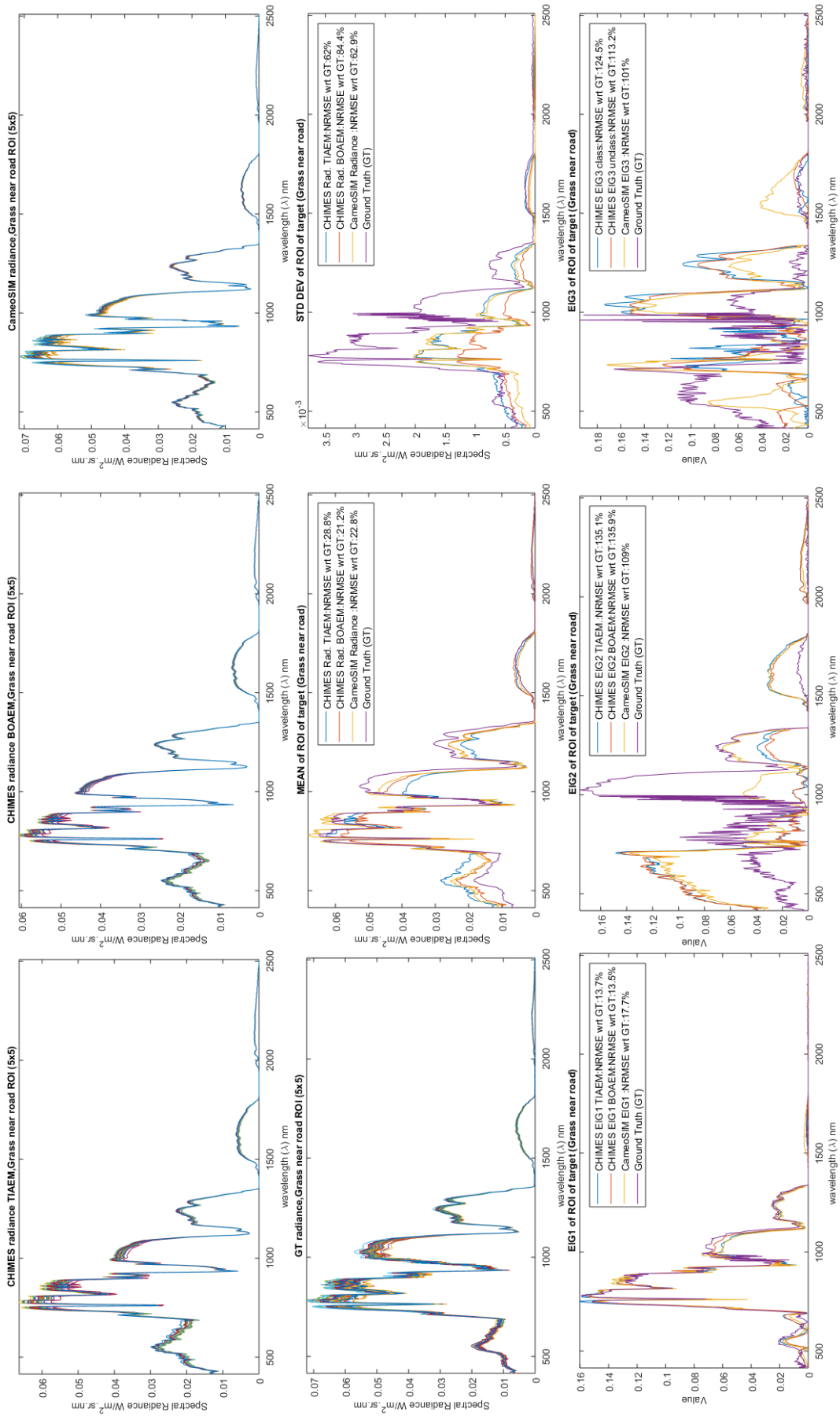


Figure A.1: **Flat Terrain:** Radiance of a ROI (5×5) for grass near road. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

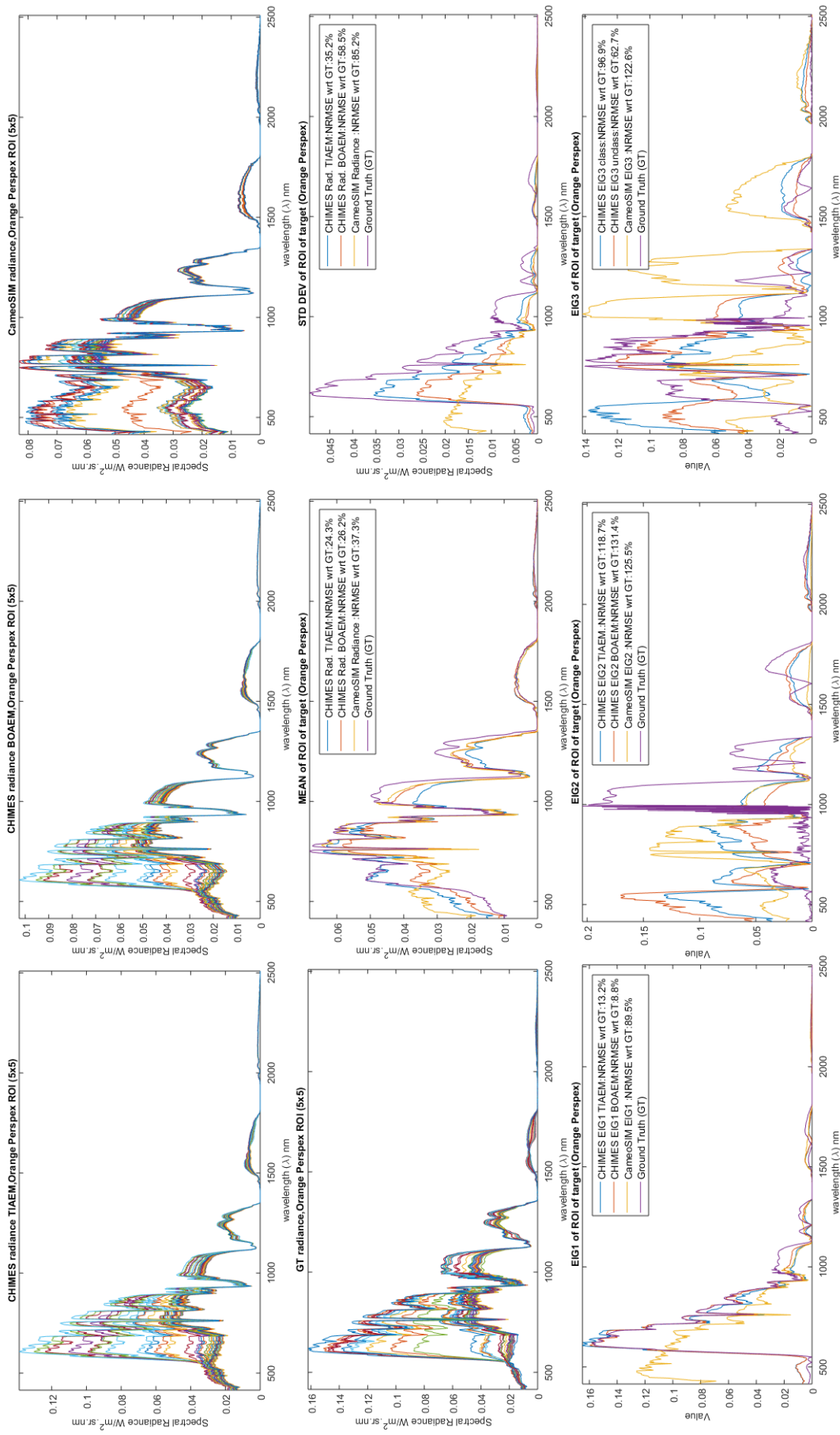


Figure A.2: **Selene Flat Terrain:** Radiance of a ROI (5 × 5) for Orange Perspex near grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

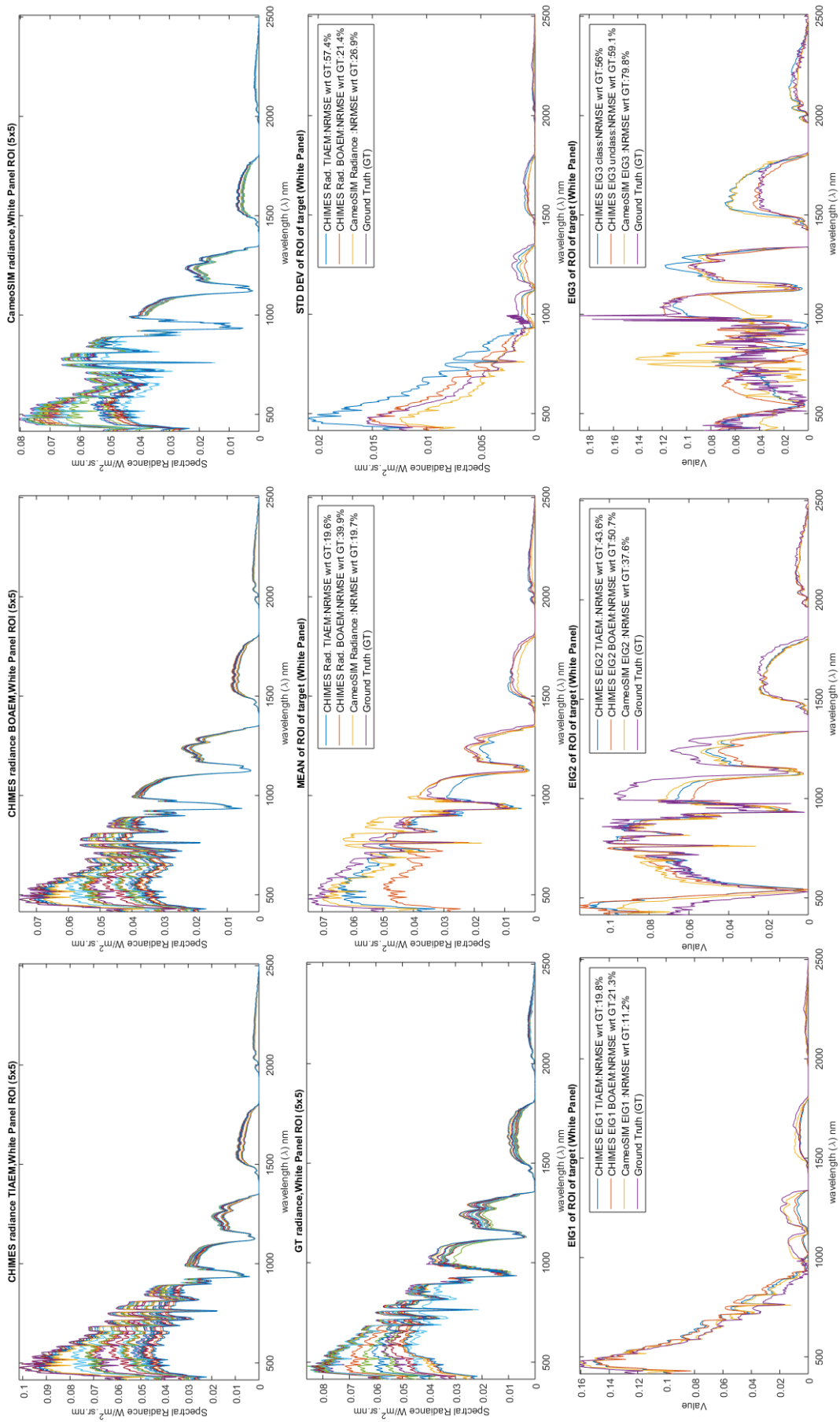


Figure A.3: **Selene Flat Terrain:** Radiance of a ROI (5×5) for White Panel. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

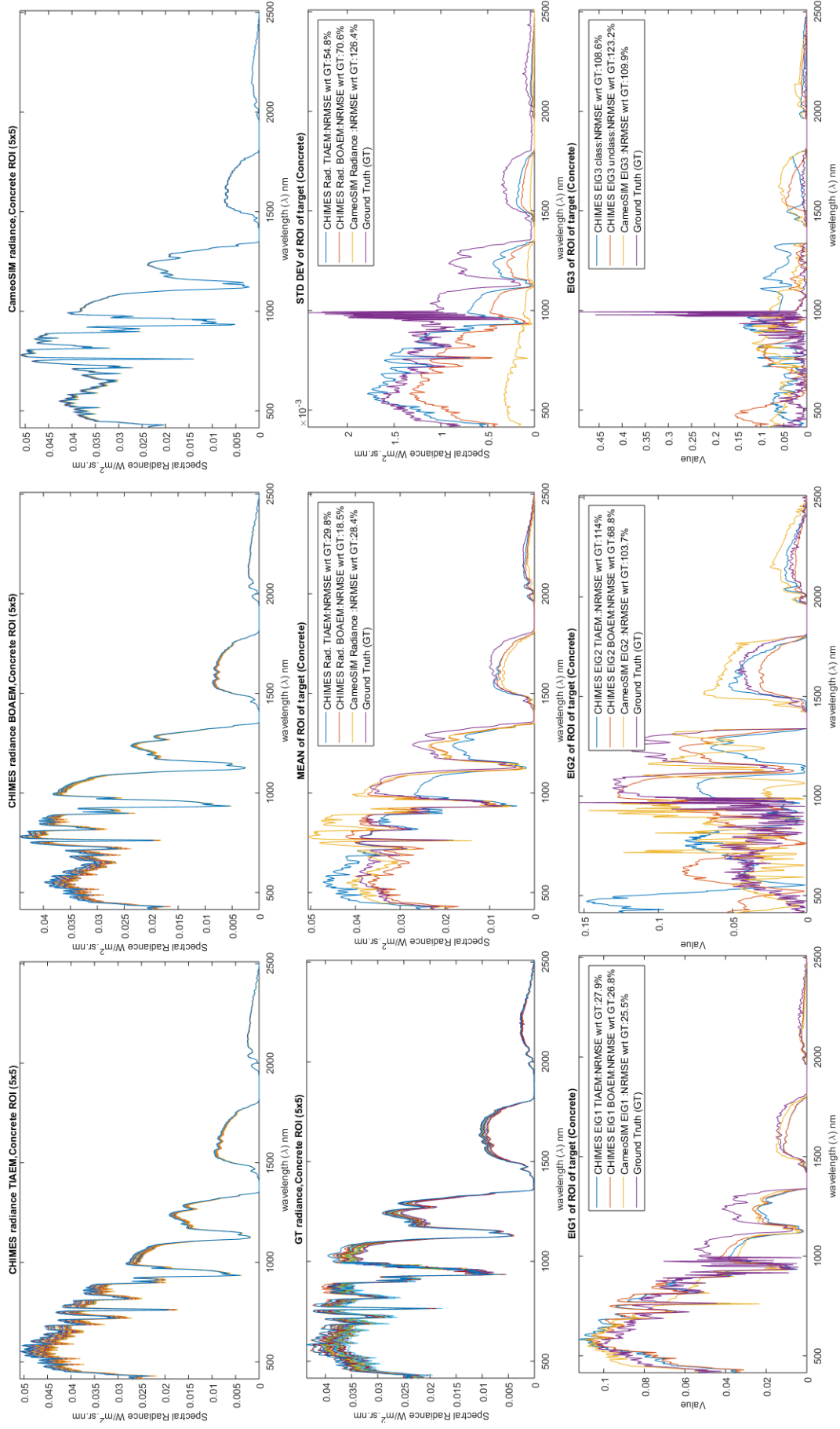


Figure A.4: **Selene Flat Terrain:** Radiance of a ROI (5×5) for Concrete. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

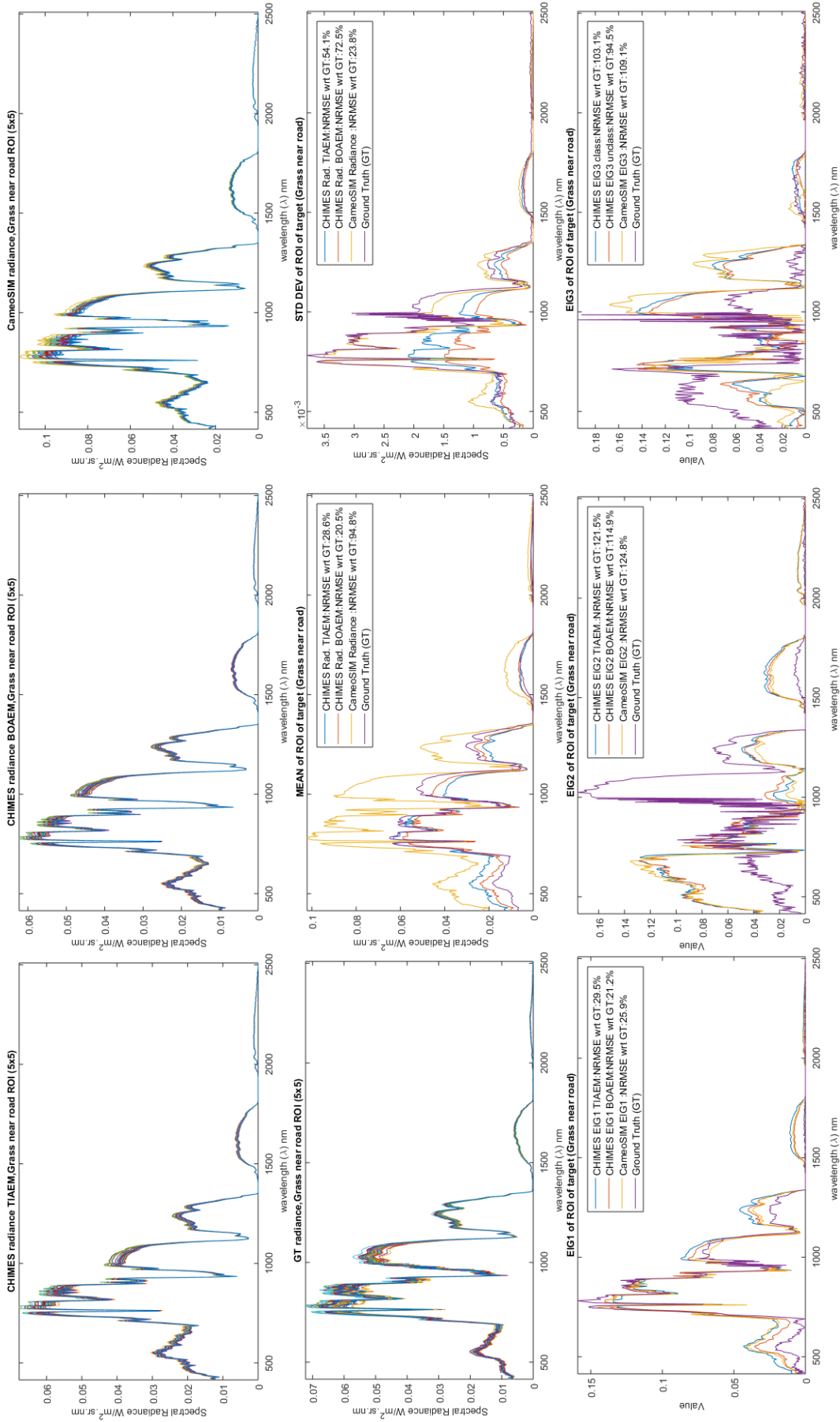


Figure A.5: **Selene DEM Terrain:** Radiance of a ROI (5×5) for Grass near road. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

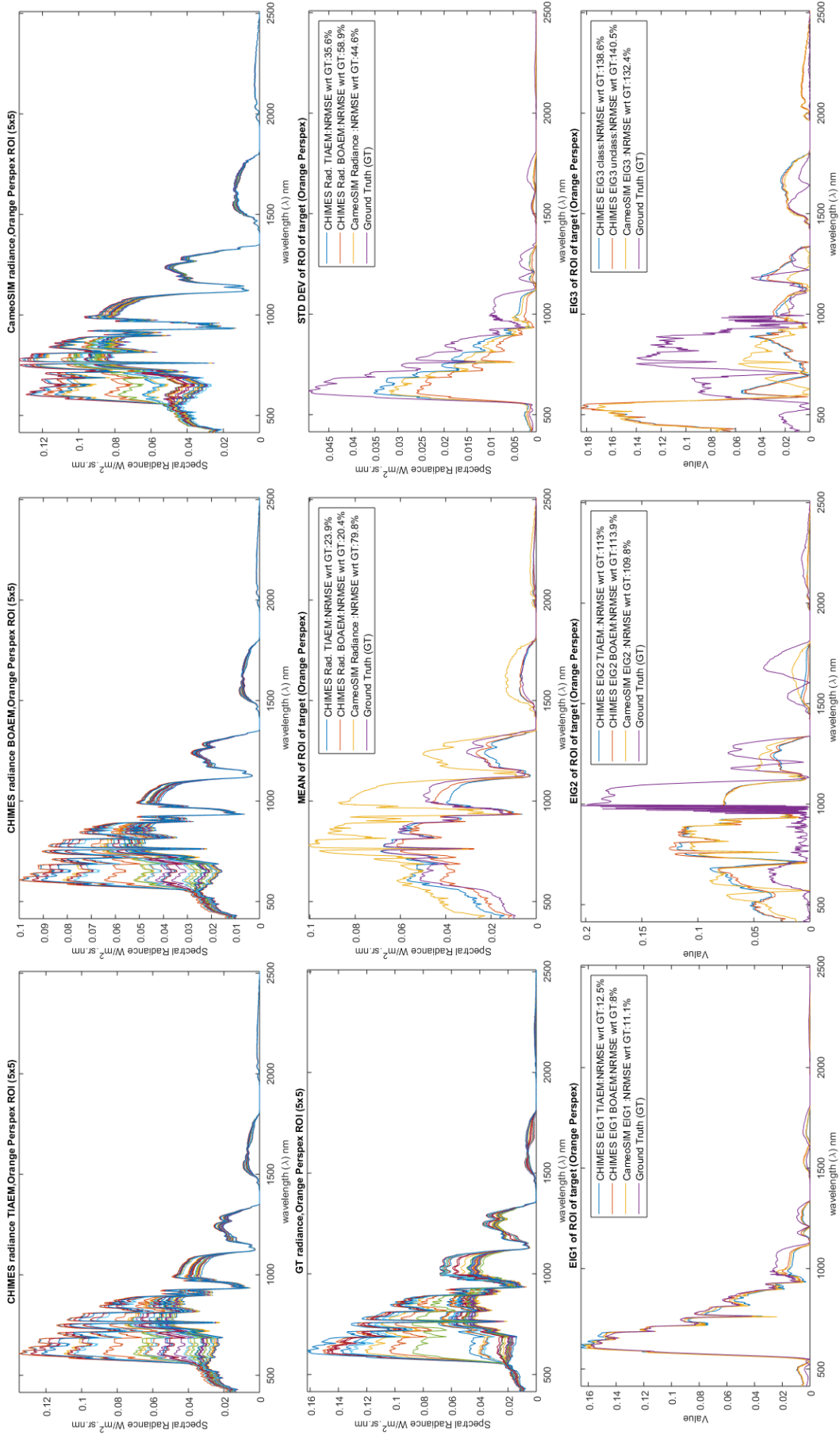


Figure A.6: **Selene DEM Terrain:** Radiance of a ROI (5×5) for Orange Perspex near grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

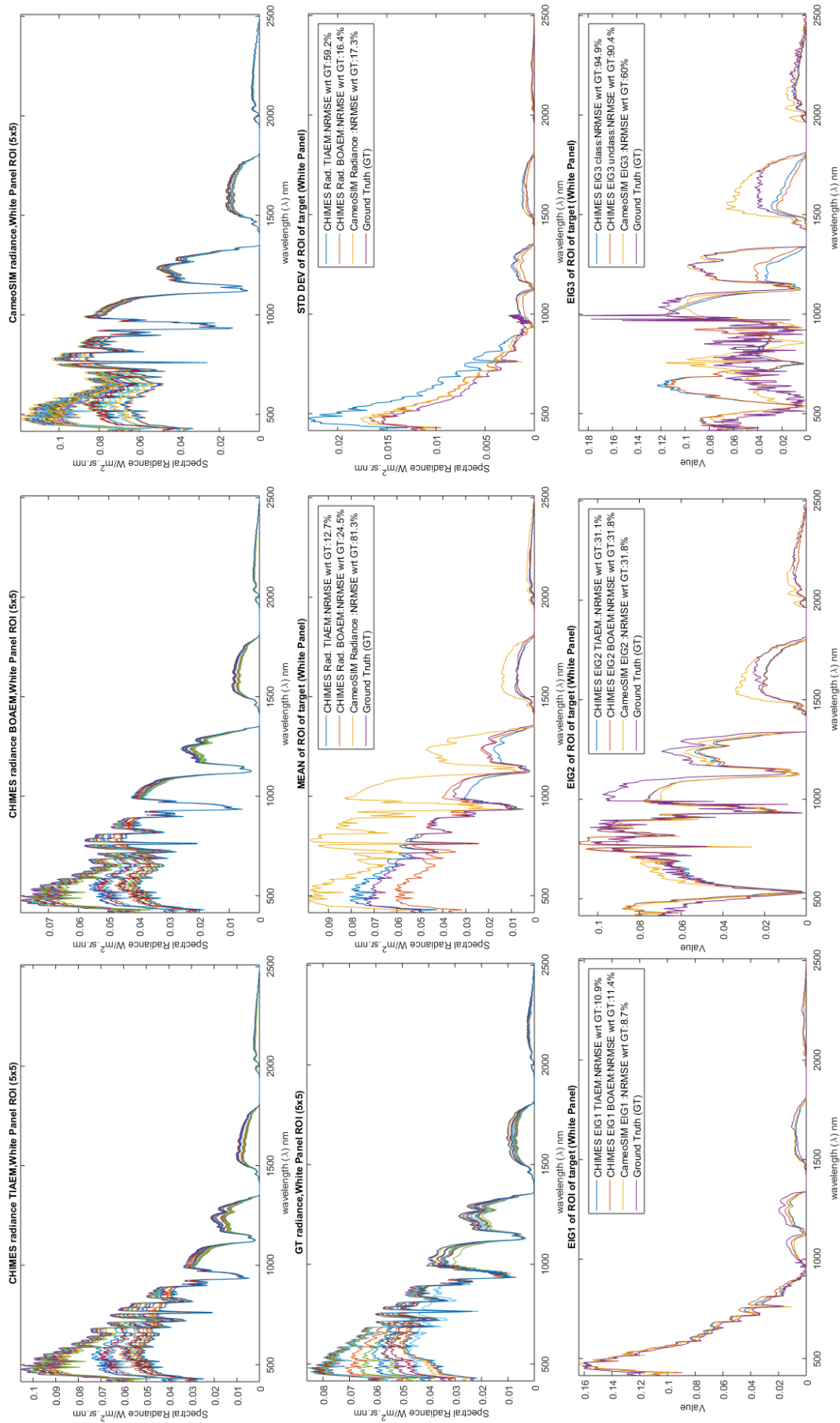


Figure A.7: **Selene DEM Terrain:** Radiance of a ROI (5 × 5) for White Panel. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

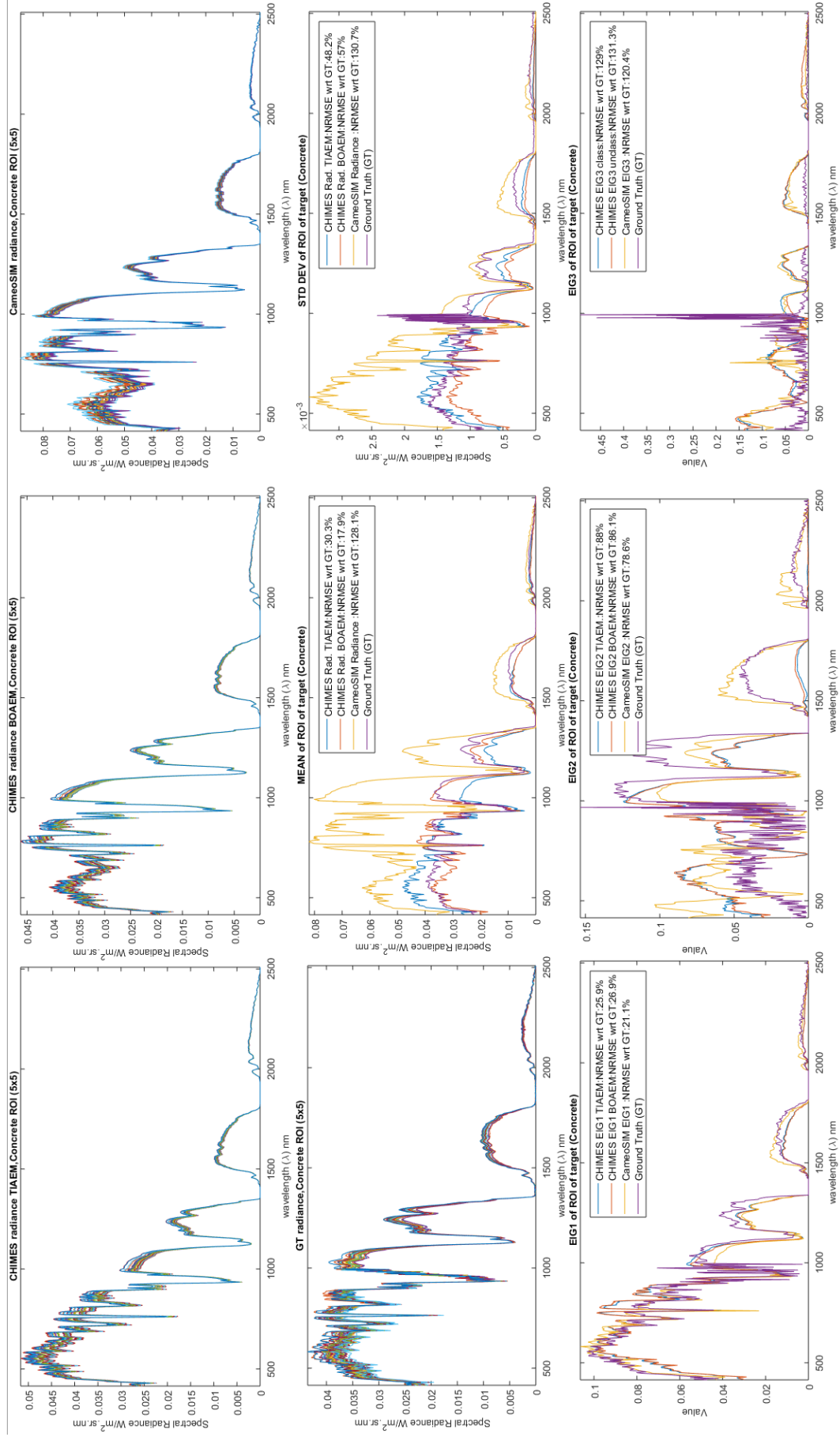


Figure A.8: **Selene DEM Terrain:** Radiance of a ROI (5×5) for Concrete. Mean, standard deviation of all simulations along with first three Eigenvalues of covariance are shown. NRMSE of entities with respect to GT is also calculated.

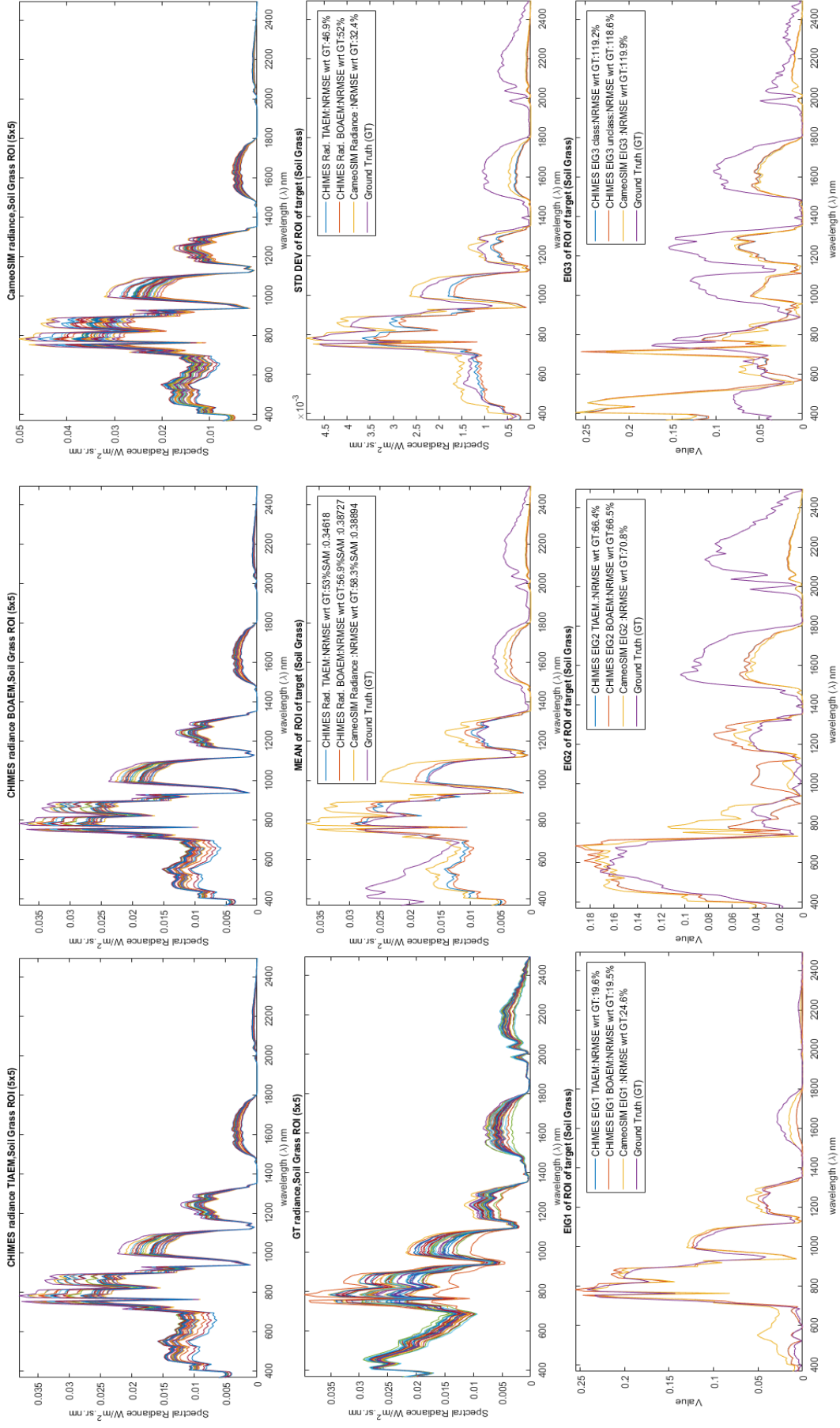


Figure A.9: **Modesto Flat Terrain**: Radiance of a ROI (5x5) for a mixture of soil and grass. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

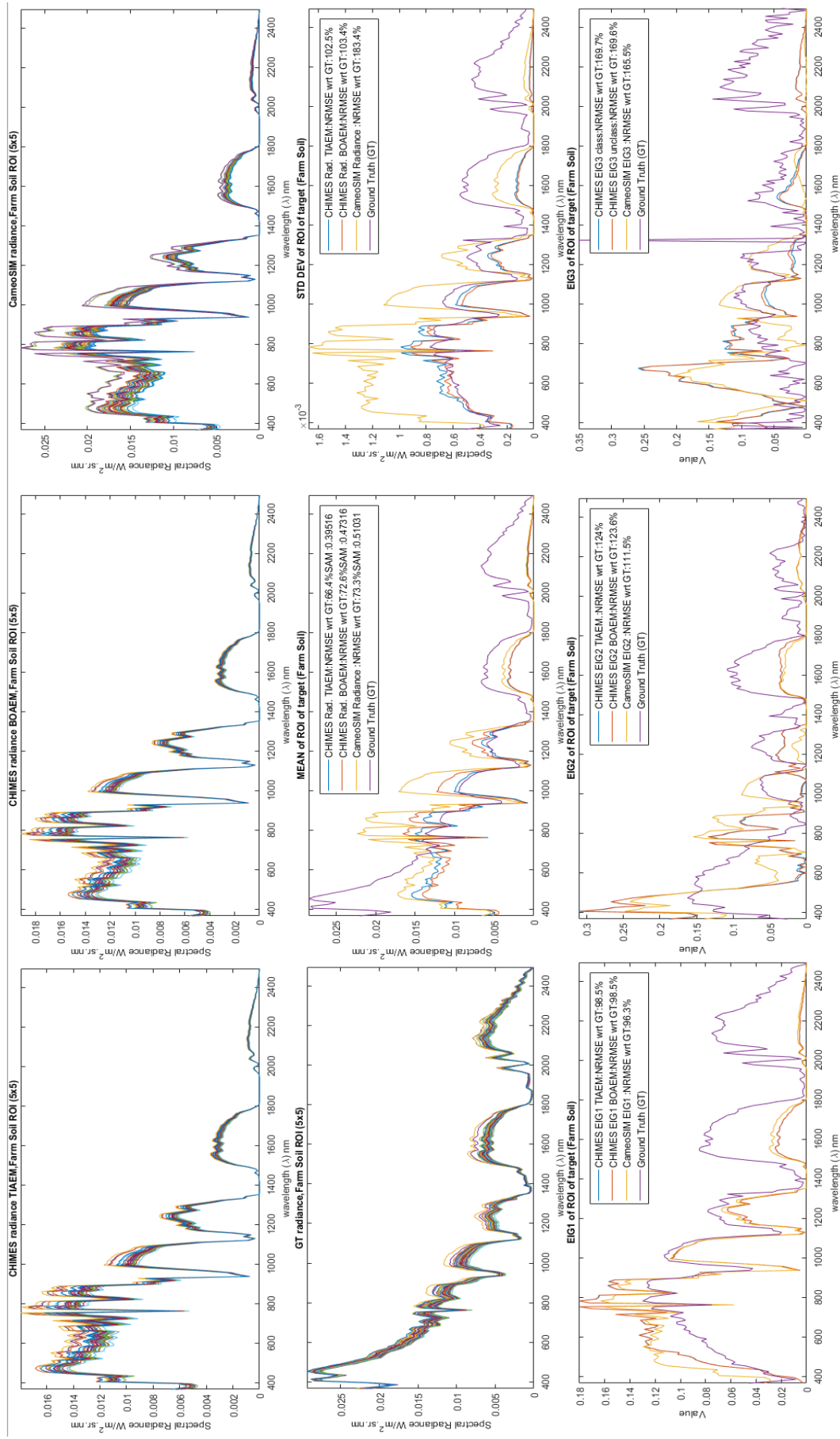


Figure A.10: **Modesto Flat Terrain:** Radiance of a ROI (5x5) for farm soil. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

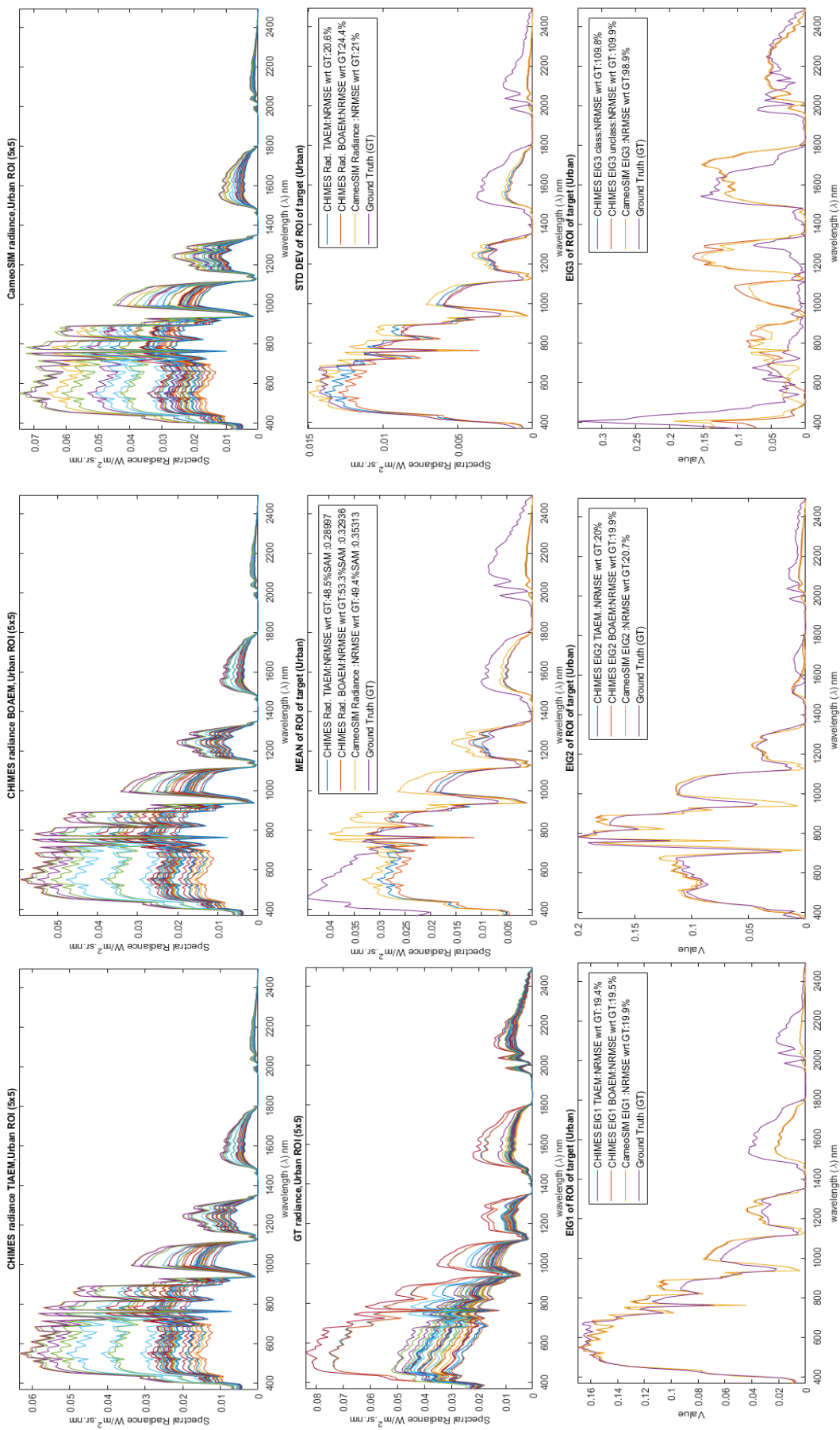


Figure A.11: **Modesto Flat Terrain:** Radiance of a ROI (5×5) for urban region of town. Mean, standard deviation of all simulations along with first three Eigenvectors of covariance are shown. NRMSE of entities with respect to GT is also calculated.

Appendix B

Source Files List

We provide a list of components developed during this project. Some generic third party tool components are also listed along with the newly developed ones. The third party tools are part of generic C++ libraries, image handling libraries and from the component in the toolkits [103],[104], [105] and [48] etc.

CHIMES		THIRD PARTY		Explanation2	
	Explanation				
chimes-pathtrace.lib	path tracer	boost_chrono-vc120-mt-1_53.dll		General C++ Library	
chimes-kernel.lib	kernel	boost_chrono-vc120-mt-1_53.lib		General C++ Library	
chimes-accelerator.lib	hardware offloader	boost_filesystem-vc120-mt-1_53.dll		General C++ Library	
chimes-render.lib	renderer main	boost_filesystem-vc120-mt-1_53.lib		General C++ Library	
texture_interp.dll	Chimes_self_explanatory	boost_python34-vc120-mt-1_53.lib		General C++ Library	
adjacency.dll	Chimes_self_explanatory	boost_system-vc120-mt-1_53.dll		General C++ Library	
area_light.dll	Chimes_self_explanatory	boost_thread-vc120-mt-1_53.dll		General C++ Library	
bitmap.dll	Chimes_self_explanatory	glew32mx.dll		Image	
boae.m.dll	Chimes_self_explanatory	glew32mx.lib		Image	
difftrans.dll	Chimes_self_explanatory	Half.dll		Image	
diffuse.dll	Chimes_self_explanatory	Half.lib		Image	
fog_medium.dll	Chimes_self_explanatory	lex.dll		reading file	
fov.dll	Chimes_self_explanatory	lex.lib		reading file	
gaussian_recons.dll	Chimes_self_explanatory	ilmimf.dll		computing	
gridtexture.dll	Chimes_self_explanatory	ilmimf.lib		computing	
hammersley.dll	Chimes_self_explanatory	lmThread.dll		computing	
heightfield.dll	Chimes_self_explanatory	lmThread.lib		computing	
hg.dll	Chimes_self_explanatory	lmath.dll		computing	
homog_medium.dll	Chimes_self_explanatory	lmath.lib		computing	
hyperchannel.dll	Chimes_self_explanatory	jpeg.lib		image	
isotropic.dll	Lambertian	jpeg62.dll		image	
imagery.dll	Chimes_self_explanatory	libpng16.dll		image	
mie_lorentz.dll	Chimes_self_explanatory	libpng16.lib		image	
normalmap.dll	Chimes_self_explanatory	python34.lib		Python Interface	
obj.dll	object file reader	libeay32.dll		GUI	
observer_view.dll	image of the scene	libiconv2.dll		GUI	
path_tracer.dll	Chimes_self_explanatory	libintl3.dll		GUI	
phong_brdf.dll	Chimes_self_explanatory	libssl32.dll		GUI	
poisson.dll	Chimes_self_explanatory	qjpeg4.dll		GUI	
ptracer.dll	Chimes_self_explanatory	QtCore4.dll		GUI	
rayleigh.dll	Chimes_self_explanatory	QtGui4.dll		GUI	
skymap.dll	Chimes_self_explanatory	qtiff4.dll		GUI	
spherical.dll	Chimes_self_explanatory	regex2.dll		GUI	
sun.dll	Chimes_self_explanatory	xerces-c_3.lib		GUI	
skydome.dll	Chimes_self_explanatory	xerces-c_3_1.dll		GUI	
tiaem.dll	Chimes_self_explanatory	zlib.dll		Compression	
		zlib.lib		Compression	

Figure B.1: A list of library files developed in CHIMES along with the third party libraries.

Appendix C

Source Codes

C.1 Getting Modtran Data

Once tape5 is created it is input into the MODTRAN and it is executed. After MODTRAN execution is completed a tape7 file is created. Data is read from the tape7 file using the following code snippet.

```
1 function tp7_interp = get_tape71_data(fnTape7, lambda, sen_resp, fwhm
   , isDual, include_totrad, qe, altitude, interpolate)
2
3 %%
4 %Started by Usman Zahidi 16 Nov 2017
5
6 % Required:
7
8 %Parameter explanation
9
10 %fnTape7           = tape7 filename
11 %lambda            = wavelenghts of sensor
12 %sen_resp          = inband sensor response data
13 %fwhm              = FWHM
14 %isDual            = dataset type in Selene context
15 %include_totrad    = bool to indicate to include flux
16 %qe                = Quantum Efficiency of sensor
17 %altitude          = altitude of sensor
18 %interpolate       = bool: to indicate interpolation
19
20 % use case:
21 % usually called from other function
22 %%
23
```



```

60
61 if(interpolate)
62     tp7_interp = getintepolated(mod_wvl, tp7rad, lambda, sen_resp, fwhm
        , isDual, isSkyPath, include_totrad, qe);
63     tp7_interp(isinf(tp7_interp))=0;
64     tp7_interp(isnan(tp7_interp))=0;
65 else
66     tp7rad(isinf(tp7rad))=0;
67     tp7rad(isnan(tp7rad))=0;
68     tp7_interp=tp7rad;
69 end
70
71 end

```

C.2 Convolving with sensor response

```

1
2 Convolution of sensor response with MODTRAN is a key step in
  creating the atmospheric data. Following code is used to
  calculate the convolution.
3 \begin{lstlisting}
4 function interpolated = getintepolated(mod_wvl, mod_data,
        sensor_lambda, sen_resp, fwhm, isDual, isSkyPath, isCloud, qe)
5
6 %%
7 %Started by Usman Zahidi 2 Dec 2017
8
9 % Required:
10
11 %Parameter explanation
12
13 %mod_wvl           = MODTRAN high resolution wavelengths
14 %mod_data          = MODTRAN radiances
15 %sensor_lambda     = Sensor wavelengths
16 %sen_resp          = Sensor inband response
17 %fwhm              = FWHM
18 %isDual            = dataset type in Selene context
19 %isSkyPath         = Clearsky path varaible for testing
20 %isCloud           = Clearsky path varaible for testing

```



```

21 %qe           = Quantum Efficiency of sensor
22
23 % use case:
24 % usually called from other function
25 %%
26
27     mod_data=mod_data(find(unique(mod_wvl)),:);
28     mod_data=mod_data(:,2:end);
29     sz=size(sen_resp,1);
30     szc=size(mod_data,2);
31     bands=size(fwhm,1);
32     nobands=double(uint16(sz/bands));
33     bw=sensor_lambda(2:end)-sensor_lambda(1:end-1);
34     bw=[bw;bw(end)];
35
36
37     mod_data=interp1(mod_wvl,mod_data,sen_resp(:,1),'linear');
38     k=1;
39     for i=1:nobands:sz
40         %i
41         if (k==255)
42             nobands=nobands-1;
43             end
44             val=sz-i;
45             if (val<nobands)
46                 nobands=val;
47                 x=sen_resp(i:(i-1)+nobands,1);
48
49                 xo=sensor_lambda(k);
50                 sigma=fwhm(k,3)/(2*sqrt(2*log(2)));
51                 fx=(1/(sigma*sqrt(2*pi)))*exp(-(x-xo).^2./(2*sigma
52 ^2));
53                 modval=mod_data(i:(i-1)+nobands,1:end);
54                 interpolated(k,:)=(qe.dual(k)*trapz((modval).*repmat
55 (fx,1,szc))./trapz(fx));
56             else
57                 x=sen_resp(i:(i-1)+nobands,1);
58                 xo=sensor_lambda(k);
59
60                 sigma=(fwhm(k,3))/(2*sqrt(2*log(2)));
61                 fx=(1/(sigma*sqrt(2*pi)))*exp(-(x-xo).^2./(2*sigma

```

```

    ^2));
60         modval=mod_data(i:(i-1)+nobands,1:end);
61         interpolated(k,:)=(qe.dual(k)*trapz((modval).*
repmat(fx,1,szc))./trapz(fx));
62         end
63         if (k==1)
64             sen_resp_all=[x,fx];
65         else
66             sen_resp_all=[sen_resp_all;[x,fx]];
67         end
68         k=k+1;
69     end
70     if (size(interpolated,1)<size(sensor_lambda,1))
71         count=size(sensor_lambda,1)-size(interpolated,1);
72         interpolated(end+1:end+count,:)=0;
73     end
74     interpolated=[sensor_lambda,interpolated];
75 end

```

C.3 Sub-pixel / Full-pixel target planting in CameoSim Environment

Sub-pixel target planting is not supported in CameoSim environment. The following script automatically create material in CameoSim format and plants it in a sub-pixel/full-pixel fashion, in its synthetic environment.

```

1
2 function plant_targets(unique_name,image,cluster,lambda,
    target_types,linux_user_name,target_reflectance,image_offset,
    lambda_offset,rotation_factor,do_flip,gsd_x,gsd_y,
    separation_distance_px,target_weights,target_per_weight,dim_x,
    dim_y,DEMImage,terrain_type,first_loc,afgFileName,objFileName,
    targets)
3 %%
4 %Started by Usman Zahidi 16 February 2017
5
6 % Required:
7 %This code should be run in linux
8 %Add a text file 'cameosim' to /etc/profile.d folder, with one line
    text PATH=$PATH:~/opt/insys/bin

```

```

9 %if providing DEM, make sure that it is same in rotation and
    flipping as
10 %the input image
11
12 %Parameter explanation
13
14 %unique_name          = unique name to prefix material,ctx,png,sh
    files and for foldername
15 %image                = hsi image cube
16 %cluster              = classification map, required to make main
    ctx
17 %lambda               = list of wavelengths
18 %target_types         = 4 if (orange perspex 30x30, green 40x40,
    green 20x20, orange 10x10)
19 %linux_user_name      = your linux user name same as in home
    directory
20 %target_reflectance   = structure that contains signature of
    targets
21 % image_offset        = ie in example 10000
22 %lambda_offset        = 1000 if lambda is in nm
23 % rotation_factor     = 3 or other accordingly
24 % do_flip             = 1 or other accordingly
25 %gsd_x                = gsd horizontal
26 %gsd_y                = gsd vertical
27 %separation_distance_px = separation distance between targets in
    pixels
28 %target_weights        = set more than one if weight should reduce
    by 20%
29 %target_per_weight    = no of targets of one weight (abundancy)
30 %dim_x                = horizontal subpixel target dimension in
    cm. For full pixel planting the size would be overwritten by
    gsdx*100.
31 %dim_y                = vertical subpixel target dimension in cm.
    For full pixel planting the size would be overwritten by gsd_y
    *100.
32 %DEMImage             = Digital Elevation Model image. It should
    be in %AGL
33 %terrain_type         = 1 if flat surface, 2 if height field
    surface, 3 %if 3D mesh polygons based surface i.e from objfile
    generated by makeObjFromMatrix.m function (ver 2.0).Function
    requires DEMImage as input

```

```

34 %first_loc           = If user wish to input coordinate by GUI
    set it to [-1 -1] else set it to planting location
35 %agfFileName        = It requires a agf format file converted
    from obj file which is generated by makeObjFromMatrix. obj to
    agf conversion is performed by cameosim.
36
37
38 %EXAMPLE about parameter with full pixel planting:
39 %vnir
40 %plant_targets('H23VNIRSub1_01Mar',H23VnirQuacSub1.imag,
    H23VnirQuacSub1.kmeans.allcluster,H23VnirQuacSub1.lambda,4,'
    usman',H23Sub1target,1,1000,3,1,0.17,0.34,10,1,25);
41
42 %dual:
43 %plant_targets('H23Dual_27Mar_1',H23DualQuac.imag,H23DualQuac.
    kmeans.allcluster,H23DualQuac.lambda,4,'usman',H23Dualtarget
    ,10000,1000,3,1,0.7,0.7,3,1,25);
44
45 % EXAMPLE with subpixel planting
46
47 %1. FLAT terrain: with calibration panel
48 %plant_targets('H23Dual_DEM_Flat',H23DualQuac.imag(:,801:1500,:),
    H23DualQuac.kmeans.allcluster(:,801:1500),H23DualQuac.lambda,6,'
    usman',H23Dualtarget,10000,1000,3,0,0.7,0.7,5,1,5,30,30,0,1,[215
    553], '');
49
50 %2. HEIGHT FIELD:
51 %plant_targets('H23Dual_DEM_HF',H23DualQuac.imag(:,801:1500,:),
    H23DualQuac.kmeans.allcluster(:,801:1500),H23DualQuac.lambda,6,'
    usman',H23Dualtarget,10000,1000,3,0,0.7,0.7,5,1,5,30,30,
    H23DEMAGL(1:380,801:1500),2,[215 553], '');
52
53 %3. DEM
54 %plant_targets('H23Dual_Obj1_0',H23DualQuac.imag(:,801:1500,:),
    H23DualQuac.kmeans.allcluster(:,801:1500),H23DualQuac.lambda,6,'
    usman',H23Dualtarget,10000,1000,3,0,0.7,0.7,5,1,5,30,30,
    H23DEMAGL(1:380,801:1500),3,[215 553],'H23Dual_Obj1.0.agf');
55 %create H23Dual_Obj1.0.agf file using makeObjFromDEM and then
    import it into CS as cgf and then convert it to agf. You can
    select different scales in makeObjFromDEM
56

```

```
57
58 %How to use:
59 %It will ask about target size, location and target type
60 %After running the code a folder is generated in working folder
    with unique_name given
61 %Create /ctx folder on linux desktop.
62 %Go in generated folder and run .sh in terminal
63 %A cgf would be generated in ctx/generated_folder_name
64 %Open cgf in CS and apply proper orientation as agf orientations
    are not converted by CS agf2cgf converter
65 %Save cgf as terrain and simulate
66
67 %A mat file with all targets is also generated with "unique_name".
    mat in the
68 %folder created, to send together or to add with results.
69 %%
70 close all;
71 brightness=2000;
72 image = rot90(image, rotation_factor);
73 if (do_flip == 1)
74     image = fliplr(image);
75 end
76
77 if (DEMImage~=0)
78 DEMImage=rot90(DEMImage, rotation_factor);
79 if (do_flip == 1)
80     DEMImage = fliplr(DEMImage);
81 end
82 end
83
84
85
86 [image_h,image_w,~]=size(image);
87 cluster = rot90(cluster, rotation_factor);
88
89 if (do_flip == 1)
90     cluster = fliplr(cluster);
91 end
92 [im_h,im_w,~]=size(image);
93 image = image./image_offset;
94 lambda = lambda./lambda_offset;
```

```

95
96 rgb_image =makeRGBimage(image,lambda);
97 all_targets=zeros(1,4);
98 targetsNumber=1;
99 target_details=cell(2,2);
100
101 first_loc_x=first_loc(1,1);
102 first_loc_y=first_loc(1,2);
103 iterations=1;
104 copy_unique_name=unique_name;
105 if (isempty(targets))
106 for i=1:target_types
107     [h,~]=size(all_targets);
108     clear target_signature;
109     isOP = input('\nEnter Target Type? (1.OrangePespex 2.GreenTile
110 3.WhitePanel)\n');
111     enable_subpixel_planting = input('\nDo you want to do subpixel
112 planting (enter 1 if yes 2 if no)\n');
113     if (enable_subpixel_planting==1)
114         sp_offset_x= input('\nEnter subpixel x_offset from bottom
115 left.Should be (target x_dimension in m + offset <= GSD_x) and
116 >=0: ');
117         sp_offset_y= input('\nEnter subpixel y_offset from bottom
118 left.Should be (target y_dimension in m + offset <= GSD_y) and
119 >=0: ');
120         iterations=2;
121         adjacent_planting=5; %plant four adjacent panels and
122 including itself it becomes 5. Adjacent panels are planted to
123 enable user to view subpixel location within pixel
124     else
125         sp_offset_x= 0; %subpixel offset in x
126         sp_offset_y= 0; %subpixel offset in y
127         adjacent_planting=1;
128         iterations=1;
129     end
130     if (isOP == 1)
131         target_material=1;
132         target_details{target_material,1}='orange_perspex_';
133         target_details{target_material,2}=num2str(target_material);
134     elseif(isOP == 2)
135         target_material=2;

```

```

128     target_details{target_material,1}='green_carpet_';
129     target_details{target_material,2}=num2str(target_material);
130 elseif(isOP == 3)
131     target_material=3;
132     target_details{target_material,1}='white_panels_';
133     target_details{target_material,2}=num2str(target_material);
134 end
135
136 if (first_loc_x<0 || first_loc_y<0)
137     for i=1:adjacent_planting
138         if (i==1)
139             [targets,~,~,first_loc_x,first_loc_y] =
makeAutoPlantTargets(rgb_image,gsd_x*100,gsd_y*100,
separation_distance_px,target_weights,target_per_weight,
sp_offset_x,sp_offset_y,enable_subpixel_planting,-1,-1,dim_x,
dim_y);
140             targets(:,5)=target_material;
141
142             %inject additional targets in von-neumann pixel
neighbourhood
143             %to verify the position of subpixel targets
144             elseif (i==2)
145                 [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
gsd_y*100,separation_distance_px,target_weights,
target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x-1,
first_loc_y,gsd_x*100,gsd_y*100);
146                 ntargets(:,5)=3;
147                 targets=[targets;ntargets];
148
149             elseif (i==3)
150                 [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
gsd_y*100,separation_distance_px,target_weights,
target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x+1,
first_loc_y,gsd_x*100,gsd_y*100);
151                 ntargets(:,5)=3;
152                 targets=[targets;ntargets];
153
154             elseif (i==4)
155                 [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
gsd_y*100,separation_distance_px,target_weights,
target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x,

```

```

156     first_loc_y-1,gsd_x*100,gsd_y*100);
157         ntargets(:,5)=3;
158         targets=[targets;ntargets];
159
160         elseif (i==5)
161             [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
162             gsd_y*100,separation_distance_px,target_weights,
163             target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x,
164             first_loc_y+1,gsd_x*100,gsd_y*100);
165             ntargets(:,5)=3;
166             targets=[targets;ntargets];
167
168         end
169     end
170     first_loc_y=first_loc_y+separation_distance_px;
171 else
172     % it adds y even to a non -1,-1 coordinates and therefore
173     % starts
174     % from there
175
176     %if (adjacent_planting>1)
177         %first_loc_y=first_loc_y+separation_distance_px;
178     %else
179         % first_loc_x=-1;
180         % first_loc_y=-1;
181     %end
182     for i=1:adjacent_planting
183         if (i==1)
184             [targets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
185             gsd_y*100,separation_distance_px,target_weights,
186             target_per_weight,sp_offset_x,sp_offset_y,
187             enable_subpixel_planting,first_loc_x,first_loc_y,dim_x,dim_y);
188             targets(:,5)=target_material;
189
190             %inject additional targets in von-neumann pixel
191             neighbourhood
192             %to verify the position of subpixel targets
193             elseif (i==2)
194                 [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
195                 gsd_y*100,separation_distance_px,target_weights,
196                 target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x-1,

```



```

186     first_loc_y,gsd_x*100,gsd_y*100);
187         ntargets(:,5)=3;
188         targets=[targets;ntargets];
189
190         elseif (i==3)
191             [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
192             gsd_y*100,separation_distance_px,target_weights,
193             target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x+1,
194             first_loc_y,gsd_x*100,gsd_y*100);
195             ntargets(:,5)=3;
196             targets=[targets;ntargets];
197
198             elseif (i==4)
199                 [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
200                 gsd_y*100,separation_distance_px,target_weights,
201                 target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x,
202                 first_loc_y-1,gsd_x*100,gsd_y*100);
203                 ntargets(:,5)=3;
204                 targets=[targets;ntargets];
205
206                 elseif (i==5)
207                     [ntargets] = makeAutoPlantTargets(rgb_image,gsd_x*100,
208                     gsd_y*100,separation_distance_px,target_weights,
209                     target_per_weight,sp_offset_x,sp_offset_y,0,first_loc_x,
210                     first_loc_y+1,gsd_x*100,gsd_y*100);
211                     ntargets(:,5)=3;
212                     targets=[targets;ntargets];
213                 end
214             end
215             first_loc_y=first_loc_y+separation_distance_px;
216         end
217
218     targets(:,1)=targetsNumber:targetsNumber-1+numel(targets(:,1));
219     targetsNumber=targets(end,1)+1;
220     if(h==1)
221         all_targets = targets;
222     else
223         all_targets = [all_targets;targets];
224     end
225     all_targets(:,1)=1:numel(all_targets(:,1));
226 end

```

```

217 else
218     all_targets=targets;
219     if (targets(1,10)==1)
220         iterations=2;
221         adjacent_planting=5;
222     elseif (targets(1,10)==2)
223         iterations=1;
224         adjacent_planting=1;
225     end
226
227 end
228
229
230 for gen_count=1:iterations
231
232 if (gen_count==1 && iterations>1)
233     suffix='';
234     copy_all_targets=all_targets;
235     unique_name=copy_unique_name;
236     all_targets(find(all_targets(:,10)==0),:)=[];
237     all_targets(find(all_targets(:,10)==2),2)=all_targets(find(
all_targets(:,10)==2),2)+1;
238     all_targets(find(all_targets(:,10)==2),3)=all_targets(find(
all_targets(:,10)==2),3)+1;
239 elseif(iterations>1)
240     suffix='_withRefPanels';
241     unique_name=strcat(unique_name,suffix);
242     all_targets=copy_all_targets;
243     all_targets(find(all_targets(:,10)==2),2)=all_targets(find(
all_targets(:,10)==2),2)+1;
244     all_targets(find(all_targets(:,10)==2),3)=all_targets(find(
all_targets(:,10)==2),3)+1;
245 end
246 all_targets(:,1)=1:numel(all_targets(:,1));
247 %copy folder to linux
248
249 injectTargets(unique_name,image, target_reflectance,all_targets,
lambda,linux_user_name);
250 %add commands to copy material files and terrain ctx files
251 linux_path=strcat('/home/',linux_user_name,'/Desktop/ctx/',
unique_name);

```

```

252 mat_file_search_token = strcat(unique_name, '_class_mat_');
253 shText = fileread(strcat(unique_name, '/', '1_', unique_name, '
    _script.sh'));
254 shell_fileID = fopen(strcat(unique_name, '/', '1_', unique_name, '
    _script.sh'), 'w');
255 ctx_file_name = strcat(unique_name, '_ctx_file');
256 %cp -r /usr/local/fromdownload /usr/local/download
257 copyStr=['cp ', ' -r ', strcat(pwd, '/', unique_name), strcat(' /home/
    ', linux_user_name, '/Desktop/ctx\n')];
258 %copyStr=strcat(strToken,);
259 shText= strcat(copyStr, shText);
260 fprintf(shell_fileID, shText);
261 fprintf(shell_fileID, '\ncs-ctxmake-6.5 -o %s/%s.ctx %s/%s.txt\n',
    linux_path, ctx_file_name, linux_path, ctx_file_name);
262 fprintf(shell_fileID, 'sudo cp %s/%s* /opt/insys/share/comeosim
    /6.5/materials\n', linux_path, mat_file_search_token);
263 %cs-agf2cgf -o /home/usman/Desktop/ctx/H23Dual_Obj0_4/
    H23Dual_Obj0_4.cgf /home/usman/Desktop/ctx/H23Dual_Obj0_4/
    H23Dual_Obj0_4.agf
264 fprintf(shell_fileID, 'cs-agf2cgf -o %s.cgf %s.agf\n', strcat(
    linux_path, '/', unique_name), strcat(linux_path, '/', unique_name));
265 fclose(shell_fileID);
266
267 %create agf files
268 if (terrain_type==1)
269     create_agf_file_from_targets(unique_name, strcat(unique_name, '/')
    , unique_name), image_w, image_h, linux_user_name, all_targets,
    gsd_x, gsd_y, 0, terrain_type, '');
270 elseif (terrain_type==2)
271     create_agf_file_from_targets(unique_name, strcat(unique_name, '/')
    , unique_name), image_w, image_h, linux_user_name, all_targets,
    gsd_x, gsd_y, DEMImage, terrain_type, afgFileName);
272 elseif (terrain_type==3)
273     all_targets = create_agf_file_from_targets(unique_name, strcat(
    unique_name, '/', unique_name), image_w, image_h, linux_user_name,
    all_targets, gsd_x, gsd_y, DEMImage, terrain_type, afgFileName);
274 end
275 %adjust target offsets
276 all_targets(find(all_targets(:,10)==2), 2)=all_targets(find(
    all_targets(:,10)==2), 2)-1;

```

```

277 all_targets(find(all_targets(:,10)==2),3)=all_targets(find(
    all_targets(:,10)==2),3)-1;
278 %create materials for main ctx file
279 create_materials_from_classification(unique_name, cluster, image,
    lambda,linux_user_name);
280 %save targets in mat file
281 result.targets=all_targets;
282 result.target_map = target_injection_map(all_targets, im_w, im_h,
    brightness)';
283 figure(1);imagesc(result.target_map);axis image;
284 %after show set brightness back
285 result.target_map = result.target_map/brightness;
286 result.target_details=target_details;
287 %save mat file
288 save(strcat(unique_name,'/',unique_name,'.mat'),'result', '-v7.3');
289 end
290 %to verify the rotaion,flip and target map of image,see and confirm
    .
291 figure(2);imagesc(makeRGBimage(image,lambda));axis image;
292 figure(3);imagesc(cluster);axis image;

```

C.4 CHIMES Codes

Here we list some of the CHIMES codes such as sky creation, phase function model, sun and sky shine calculation and finally upwelled scattered radiance calculation in the renderer.

```

1 // Function in Sky class which creates a bitmap from MODIRAN
    interpolated data for each pixel of the bitmap. Width of bitmap
    represents the azimuth and height is zenith. Sun is positioned
    according to user input of longitude ,latitude , date and time of day
    . Bitmap is converted into spherical coordinate to create a skydome
    .
2
3 // Loaded when a project definition file is loaded
4 SpectrumVector CreateSky(Float theta , Float azimuth , Float R,Float
    resolution)
5 {
6 ref<Bitmap> bitmap = new Bitmap(PFORMAT, Bitmap::EFloat ,
7     Vector2i(resolution , resolution / 2));

```

```

8
9 Point2 factor((2 * M_PI) / bitmap->getWidth(),
10 M_PI / bitmap->getHeight());
11
12
13 SpectrumVector *data = (SpectrumVector *)bitmap->getFloatData();
14
15 SpectrumVector groundAlbedo = SpectrumVector::
    GetInterpolatedSpectrumVector(SpectrumVector::m_projectID, int(
    DataAccess::EMOD::EDownwell));
16
17 for (int y = 0; y<bitmap->getHeight(); ++y)
18 {
19     Float theta = (y + .5f) * factor.y;
20     SpectrumVector *target = data + y * bitmap->getWidth();
21
22     for (int x = 0; x<bitmap->getWidth(); ++x)
23     {
24         Float phi = (x + .5f) * factor.x;
25
26         RayDiff ray(Point(0.0f, 0.0f, 0.0f), toSphere(SphericalCoord(theta,
            phi)), 100.0f);
27         RayDiff raySolar(Point(0.0f, 0.0f, 0.0f), toSphere(SphericalCoord(
            SpectrumVector::solarZenith, SpectrumVector::azimuth)), 100.0f);
28         SpectrumVector T1 = SpectrumVector::GetInterpolatedSpectrumVector
            (-1, int(DataAccess::EMOD::ETrans1));
29         SpectrumVector rayleighScat = getRayleighPhase(raySolar.d, ray.d);
30         SpectrumVector skyRad = getSunRadiance(fromSphere(raySolar.d).
            elevation, phi, 100.0f);
31         SpectrumVector val = SpectrumVector(0.0);
32         val = getSkyShine(theta, phi, SpectrumVector::altitude) *mieScat;
33         val.condition();
34         *target++ = val;
35     }
36 }
37 }
38 //Get sun radiance from MODTRAN interpolation database
39 SpectrumVector getSunRadiance(Float theta, Float azimuth, Float R)
40 {
41     SpectrumVector TOA = SpectrumVector::GetInterpolatedSpectrumVector(
        SpectrumVector::m_projectID, int(DataAccess::EMOD::ETOA, R / sin(

```

```

    theta), azimuth, theta));
42 SpectrumVector T1 = SpectrumVector::GetInterpolatedSpectrumVector(
    SpectrumVector::m_projectID, int(DataAccess::EMOD::ETrans2), R /
    sin(theta), azimuth, theta);
43 SpectrumVector lightShine = TOA * cos(theta)*T1;
44 lightShine.clampNegative();
45 return lightShine;
46 }
47 //Get sky radiance from MODTRAN interpolation database
48 SpectrumVector getSkyShine(Float theta, Float azimuth, Float R)
49 {
50     int projID = SpectrumVector::m_projectID;
51     SpectrumVector skyShine(0.0f);
52     SpectrumVector mulScat = SpectrumVector::
53         GetInterpolatedSpectrumVector(SpectrumVector::m_projectID,
54         DataAccess::EMOD::EMultScat, R, azimuth, theta);
55     SpectrumVector downWelledRadiance = SpectrumVector::
56         GetInterpolatedSpectrumVector(SpectrumVector::m_projectID,
57         DataAccess::EMOD::EDownwell, R, azimuth, theta);
58     if (mulScat.isZero())
59     {
60         SpectrumVector TOA = SpectrumVector::GetInterpolatedSpectrumVector(
61         SpectrumVector::m_projectID, int(DataAccess::EMOD::ETOA));
62         SpectrumVector t1 = SpectrumVector::GetInterpolatedSpectrumVector(
63         SpectrumVector::m_projectID, DataAccess::EMOD::ETrans2, R, azimuth,
64         theta);
65         skyShine = (TOA*cos(theta)* t1);
66     }
67     else
68     {
69         skyShine = downWelledRadiance * M_PI; %diffuse
70     }
71     skyShine.clampNegative();
72     SpectrumVector::m_projectID = projID;
73     return skyShine;
74 }
75 //Rayleigh phase function defined in PhaseFunction class
76 SpectrumVector getRayleighPhase(Vector wsol, Vector wscat)
77 {

```

```

73   Float u = (dot(wsol, wscat) / (magnitude(wsol)*magnitude(wscat)));
74   Float delta = 0.0279f; //page 159 MODTRAN report, incorporates
      depolarizing factor
75   Float p = (3.0f / 4.0f)*((1.0f + delta) + (1.0f - delta)*u*u);
76   return SpectrumVector(p);
77 }

```

C.5 Adjacency Model Rendering

```

1 //Functions in Renderer class for computing upwelled scattered radiance
      contribution for a given ray
2 inline std::vector<int> UpwelledRadiance(Ray ray, TextureCoord uv, int R
      , int classes, std::vector<float> emitterPdf, std::vector<float>
      phasePdf, Intersection its, float localAdjacencyWeight) const
3 //localAdjacencyWeight //if passed 0.0 then BOAEM else 1.0 TIAEM
4 bool adjacency = false;
5
6 if (localAdjacencyWeight > 0.0f)
7 {
8     adjacency = true;
9     localBackgroundAlbedo = getAdjacentAlbedo(uv, R, classes);
10    groundAlbedo = ((1.0f - localAdjacencyWeight) * grassAlbedo +
      localAdjacencyWeight*localBackgroundAlbedo);
11 }
12 else
13 {
14     adjacency = false;
15     groundAlbedo = grassAlbedo;
16 }
17 /*
      =====
      */
18 /*           phase function sampling (diffuse sources)*/
19 /*
      =====
      */
20
21 Float phasePdf;
22 PhaseFunctionSamplingData phaseData(mData, -ray.d);

```

```

23 Float phaseVal = phase->sample(phaseData, phasePdf, rRec.sampler);
24
25 if (phaseVal == 0)
26 break;
27
28 Float diffuseWeight(0.0f);
29 SpectrumVector pathRadianceBlack = SpectrumVector::
    GetInterpolatedSpectrumVector(SpectrumVector::m_projectID,
    DataAccess::EMOD::EUpwell, SpectrumVector::m_maxAltitude,
30 fromSphere(ray.d).azimuth, m_nadir) / T2*throughput;
31
32 SpectrumVector pathRadianceWhite = SpectrumVector::
    GetInterpolatedSpectrumVector(SpectrumVector::m_projectID,
    DataAccess::EMOD::EDownwell, SpectrumVector::m_maxAltitude,
33 fromSphere(ray.d).azimuth, m_nadir) / T2*throughput;
34
35 SpectrumVector pathRadianceGrass = SpectrumVector::
    GetInterpolatedSpectrumVector(SpectrumVector::m_projectID,
    DataAccess::EMOD::EMultScat, SpectrumVector::m_maxAltitude,
36 fromSphere(ray.d).azimuth, m_nadir) / T2*throughput;
37
38 Float F = 0.0f;
39
40 if (!valueSky.isZero() && (rRec.type & RadianceData::
    EDirectMediumRadiance))
41 {
42
43 diffuseWeight = misWeight(phasePdf, emitterPdf);
44 Vector absN = (Vector)its.geoFrame.n;
45 absN.x = abs(absN.x);
46 Float sigma = (acos(dot(absN, Normal(1.0, 0.0, 0.0)) / (magnitude((
    Vector)absN) * magnitude(Vector(1.0, 0.0, 0.0)))));
47 F = -0.5f*cos(sigma);
48 Lsky = F*(valueSky)*bsdfWeight*directWeight;
49 Lback = (1 - F)*(valueSky)*backgroundAlbedo*directWeight;
50 }
51 pathRadianceBlack.condition();
52 pathRadianceWhite.condition();
53 SpectrumVector pathRadiance(0.0);
54 SpectrumVector interppathRadiance(0.0);
55 SpectrumVector grasspathRadiance(0.0);

```



```

56 float scaleFactor(1.0);
57
58 if (!grassAlbedo.isZero())
59 {
60     if (adjacency)
61     {
62         pathRadiance = ((SpectrumVector(1.0f) - groundAlbedo) *
63             pathRadianceBlack) + (groundAlbedo * pathRadianceWhite); //
64             vectorized linear interpolation
65     }
66     else
67     {
68         pathRadiance = pathRadianceGrass;
69     }
70 }
71
72 Lpath += pathRadiance * diffuseWeight;
73 Lsky.condition();
74 Lup.condition();
75 Lback.condition();
76
77 Lw += (Lsky + Lup) + Lback;
78 Lw.clampNegative();
79 Lw.condition();
80 Lb = Lpath;
81 Lb.clampNegative();
82 Lb.condition();
83 Li += Lw+Lb;
84 return Li;
85 }
86 // Function in Renderer for calculating the value and weight of R
87 // Equation 3.5 and 3.8 respectively
88 inline std::vector<int> getRectFromPixel(Point2i& pos, int kernelSize,
89     bool &isFullSpan) const
90 {
91     std::vector<int> roiRect;
92     int span = kernelSize / 2;
93     int leftSpan = pos.x - span;
94     int rightSpan = pos.x + span;

```

```
93  int topSpan = pos.y - span;
94  int bottomSpan = pos.y + span;
95
96  if (leftSpan <= 0)
97  {
98      roiRect.push_back(0);
99      isFullSpan = false;
100 }
101 else
102 {
103     roiRect.push_back(leftSpan);
104 }
105
106
107 if (topSpan <= 0)
108 {
109     roiRect.push_back(0);
110     isFullSpan = false;
111 }
112 else
113 {
114     roiRect.push_back(topSpan);
115 }
116
117
118 if (rightSpan > width - 1)
119 {
120     roiRect.push_back(width - 1);
121     isFullSpan = false;
122 }
123 else
124 {
125     roiRect.push_back(rightSpan);
126 }
127
128
129 if (bottomSpan > height - 1)
130 {
131     roiRect.push_back(height - 1);
132     isFullSpan = false;
133 }
```

```

134     else
135     {
136         roiRect.push_back(bottomSpan);
137     }
138
139     return roiRect;
140 }
141
142 inline std::vector<int> getRectFromPixel(Point2i& pos, int kernelSize)
143     const
144 {
145     std::vector<int> roiRect;
146
147     int span = kernelSize / 2;
148     int leftSpan = pos.x - span;
149     int rightSpan = pos.x + span;
150     int topSpan = pos.y - span;
151     int bottomSpan = pos.y + span;
152
153     roiRect.push_back(leftSpan);
154     roiRect.push_back(topSpan);
155     roiRect.push_back(rightSpan);
156     roiRect.push_back(bottomSpan);
157
158     return roiRect;
159 }
160 inline SpectrumVector getAdjacentAlbedo(const Point2 uv, const int
161     kernelSize, const int classes) const {
162
163     int x = math::floorToInt((uv.x)*width);
164     int y = math::floorToInt((uv.y)*height); // get size properly
165
166     int matID = 0;
167     Point2i pos(x, y);
168     Point2i lastFullPos(-1, -1);
169     int original_subset = 32;
170     int subset = original_subset;
171
172     SpectrumVector albedoSpectrumVector(0.0f);

```

```

173 SpectrumVector textureSpectrumVector(0.0f);
174 SpectrumVector pixelValue(0.0f);
175 bool isFullSpan = true;
176 std::vector<int> roiRect = getRectFromPixel(pos, kernelSize,
177     isFullSpan);
178 SpectrumVector backgroundAlbedo;
179 float gsd = 0.7f;
180 Point2i ajdPos;
181
182 float abundancy = 1.0f;
183 int k = 0;
184 for (int x = roiRect[0]; x < roiRect[2]; x = x + subset)
185 {
186     for (int y = roiRect[1]; y < roiRect[3]; y = y + subset)
187     {
188
189         ajdPos = Point2i(x, y);
190         if (ajdPos != pos)
191         {
192             float dist_x = gsd*gsd*(ajdPos.x - pos.x)*(ajdPos.x - pos.x);
193             float dist_y = gsd*gsd*(ajdPos.y - pos.y)*(ajdPos.y - pos.y); //
194             ;
195             float dist = sqrt(dist_x + dist_y);
196
197             if ((ajdPos.x < width) && (ajdPos.y < height) && (ajdPos.x >=
198 0) && (ajdPos.y >= 0))
199             {
200                 matID = MaterialIDfromPosition(ajdPos);
201                 textureSpectrumVector.GetSpectrumVectorByMaterialID(
202 SpectrumVector::m_projectID, matID);
203             }
204             albedoSpectrumVector += textureSpectrumVector / (dist*dist);
205             k += 1;
206         }
207     }
208 }
209 backgroundAlbedo = albedoSpectrumVector;
210 return backgroundAlbedo;

```

```
210
211 inline int MaterialIDfromPosition(Point2i pixelPos) const
212 {
213     SpectrumVector result(0.0f);
214     // fetches reflectance from db
215
216     SpectrumVector mat = m_codedTexBitmap->getPixel(pixelPos, Bitmap::
        EPixelFormat::ERGB);
217
218     int rInt = (int)(mat[0] * 255);
219     int gInt = (int)(mat[1] * 255);
220     int bInt = (int)(mat[2] * 255);
221     return ((rInt & 0x0ff) << 16) | ((gInt & 0x0ff) << 8) | (bInt & 0x0ff
        );
222 }
```