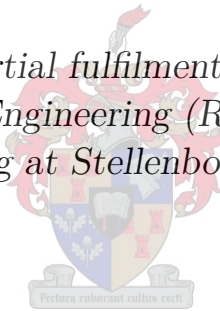


Concise Analysis and Testing of a Software Model of a Satellite Remote Sensing System used for Image Generation

by

Lynette van der Westhuizen

Thesis presented in partial fulfilment of the requirements for the degree Master of Engineering (Research) in the Faculty of Engineering at Stellenbosch University



Supervisors:

Prof. J. du Preez

Department of Electrical and Electronic Engineering

Mr. C.J. Willers

Council for Scientific and Industrial Research

December 2014

Declaration

By submitting this thesis electronically, I, Lynette van der Westhuizen, declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: December 2014

Copyright © 2014 Stellenbosch University
All rights reserved.

Abstract

Concise Analysis and Testing of a Software Model of a Satellite Remote Sensing System used for Image Generation

The capability of simulating the output image of earth observation satellite sensors is of great value, as it reduces the dependency on extensive field tests when developing, testing and calibrating satellite sensors. The aim of this study was to develop a software model to simulate the data acquisition process used by passive remote sensing satellites for the purpose of image generation. To design the software model, a comprehensive study was done of a physical real world satellite remote sensing system in order to identify and analyse the different elements of the data acquisition process. The different elements were identified as being the target, the atmosphere, the sensor and satellite, and radiation. These elements and a signature rendering equation are used to model the target-atmosphere-sensor relationship of the data acquisition process. The signature rendering equation is a mathematical model of the different solar and self-emitted thermal radiance paths that contribute to the radiance reaching the sensor. It is proposed that the software model be implemented as an additional space remote sensing application in the Optronics Sensor Simulator (OSSIM) simulation environment. The OSSIM environment provides the infrastructure and key capabilities upon which this specialist work builds. OSSIM includes a staring array sensor model, which was adapted and expanded in this study to operate as a generic satellite sensor. The OSSIM signature rendering equation was found to include all the necessary terms required to model the at-sensor radiance for a satellite sensor with the exception of an adjacency effect term. The equation was expanded in this study to include a term to describe the in-field-of-view adjacency effect due to aerosol scattering. This effect was modelled as a constant value over the sensor field of view. Models were designed to simulate across-track scanning mirrors, the satellite orbit trajectory and basic image processing for geometric discontinuities. Testing of the software model showed that all functions operated correctly within the set operating conditions and that the in-field-of-view adjacency effect can be modelled effectively by a constant value over the sensor field of view. It was concluded that the satellite remote sensing software model designed in this study accurately simulates the key features of

ABSTRACT

iii

the real world system and provides a concise and sound framework on which future functionality can be expanded.

Uittreksel

Beknopte Analise en Toetsing van 'n Sagteware Model van 'n Satelliet Afstandswaarneming Stelsel wat vir Beeld Generering Gebruik Word

Dit is nuttig om 'n sagteware program te besit wat die gegengereerde beelde van 'n satellietsensor vir aarde-waarneming kan naboots. So 'n sagteware program sal die afhanklikheid van breedvoerige veldwerktoues verminder gedurende die ontwerp, toetsing en kalibrasie fases van die ontwikkeling van 'n satellietsensor. Die doel van hierdie studie was om 'n sagteware model te ontwerp wat die dataverwerwingsproses van 'n passiewe satelliet afstandswaarnemingstelsel kan naboots, met die doel om beelde te genereer. Om die sagteware model te ontwerp het 'n omvattende studie van 'n fisiese regte wêreld satelliet afstandswaarnemingstelsel geveer, om die verskillende elemente van die dataverwerwingsproses te identifiseer en te analiseer. Die verskillende elemente is geïdentifiseer as die teiken, die atmosfeer, die sensor en satelliet, en vloed. Hierdie elemente, tesame met 'n duimdrukvergelyking, is gebruik om die teiken-atmosfeer-sensor verhouding van die dataverwerwingsproses te modelleer. Die duimdrukvergelyking is 'n wiskundige model van die verskillende voortplantingspaaie van gereflekteerde sonvloed en self-stralende termiese vloed wat bydra tot die totale vloed wat die sensor bereik. Dit is voorgestel dat die sagteware model as 'n addisionele ruimte afstandswaarnemingtoepassing in die 'Optronics sensor Simulator' (OSSIM) simulasië-omgewing geïmplementeer word. Die OSSIM simulasië-omgewing voorsien die nodige infrastruktuur en belangrike funksies waarop hierdie spesialis werk gebou kan word. OSSIM het 'n starende-skikking sensor model wat in hierdie studie aangepas is en uitgebrei is om as 'n generiese satellietsensor te funksioneer. Die OSSIM duimdrukvergelyking bevat al die nodige radiometriese terme, behalwe 'n nabyheids-verstrooiing term, om die vloed by die satellietsensor te modelleer. Die duimdrukvergelyking is uitgebrei in hierdie studie om 'n term in te sluit wat die verstrooiing van vloed vanaf naby-geleë voorwerpe, as gevolg van aerosol verstrooiing, kan beskryf. Die nabyheids-verstrooiing is gemodelleer as 'n konstante waarde oor die sigveld van die sensor. Modelle is ontwerp om die beweging van oor-baan skandering-spieëls en die satelliet wentelbaan trajek te bereken. 'n Basiese beeldverwerkings model is ook

ontwerp om diskontinuiteite in geometriese vorms in die sensor beelde reg te stel. Toetsing van die sagteware model het gewys dat al die funksies korrek gefunksioneer het binne die limiete van die vasgestelde operasionele voorwaardes. Die toets resultate het ook bewys dat die in-sig-veld nabyheids-verstrooiing akkuraat gemodeleer kan word as 'n konstante waarde oor die sensor sigveld. Daar is tot die gevolgtrekking gekom dat die satelliet afstandswaarneming sagteware model wat in hierdie studie ontwerp is al die belangrikste kenmerke van die werklike wêreld stelsel kan simuleer. Die model vorm 'n beknopte en stewige raamwerk waarop toekomstige werk uitgebrei kan word.

Acknowledgements

I would like to express my sincere gratitude to Mr. C.J. Willers and Prof. J. du Preez for being my supervisors throughout the time it took me to complete this research and write my thesis. I would like to thank Mr. D. Griffith for his technical advice and Dr. D. Bezuidenhout and the CSIR (DPSS) for providing me with the opportunity to do my master's degree.

Dedications

This thesis is dedicated to all my loved ones to whom I am very grateful for all their support and prayers throughout my studies.

Contents

Declaration	i
Abstract	ii
Uittreksel	iv
Acknowledgements	vi
Dedications	vii
Contents	viii
List of Figures	xii
List of Tables	xv
Nomenclature	xvi
1 Introductory Summary	1
1.1 Study topic and motivation	1
1.2 Study objectives	2
1.3 Earth observation using a passive satellite remote sensing system	3
1.4 Overview of the work	5
1.5 Contributions and future work	17
2 Earth Observation Using Satellite Remote Sensing	18
2.1 Introduction	18
2.2 The satellite remote sensing system	18
2.3 Existing remote sensing simulation models	34
2.4 Conclusion	34
3 Design of a Satellite Remote Sensing Software Model	35
3.1 Introduction	35
3.2 Functional diagram for the sensor image capturing process	36

3.3	Motivation for simulating the satellite image capturing process	39
3.4	Conclusion	40
4	The Signature Rendering Equation	41
4.1	Introduction	41
4.2	The satellite signature rendering equation	42
4.3	Including the adjacency effect in the signature rendering equation	46
4.4	Implementing the signature rendering equation	49
4.5	Conclusion	49
5	The Atmosphere	51
5.1	Introduction	51
5.2	Radiative transfer codes	51
5.3	Description of the implemented atmosphere	52
5.4	Conclusion	59
6	The Satellite Sensor	60
6.1	Introduction	60
6.2	A generic satellite sensor model	61
6.3	Implemented generic satellite sensor model	61
6.4	Conclusion	73
7	The Satellite Swath Scanning	74
7.1	Introduction	74
7.2	The generic satellite scanning model	75
7.3	Implementing the satellite along-track scanning configuration model	75
7.4	Implementing the satellite across-track scanning configuration model	76
7.5	Conclusion	78
8	The Satellite Trajectory	79
8.1	Introduction	79
8.2	The satellite orbit calculation model	80
8.3	Implementing the satellite orbit calculation model	81
8.4	Conclusion	89
9	Image Pre-Processing for Geometric Discontinuities	90
9.1	Introduction	90
9.2	Basic image pre-processing model	90
9.3	Implementing the staggered line correction algorithm	91
9.4	Implementing the alternating scan line correction algorithm	93
9.5	Conclusion	95

10 Implementation Issues and Limitations	96
10.1 Introduction	96
10.2 Limitations of the real world scenes and targets	96
10.3 Implementation issues of the atmospheric aerosol scattering point spread function	97
10.4 Limitations in the sensor charged coupled device focal plane layout . . .	98
10.5 Limitations in the model due to time delay integration	99
10.6 Limitations in the satellite orbit	99
10.7 Conclusion	99
11 Validation of the Model	100
11.1 Introduction	100
11.2 Atmospheric aerosol scattering for the in-field-of-view adjacency effect .	102
11.3 Staggered detector focal plane layout	108
11.4 Shifted detector focal plane layout	111
11.5 Noise equivalent reflectance	114
11.6 Analogue-to-digital conversion	117
11.7 Across-track scanning using a rotating scan mirror	119
11.8 Across-track scanning using an oscillating scan mirror	123
11.9 Image appearance due to orbit direction and earth rotation	127
11.10 Satellite full circular orbit	131
11.11 Satellite orbit trajectory between coordinates	135
11.12 Staggered detector layout image correction	138
11.13 Oscillating scan mirror image correction	142
11.14 Conclusion	144
12 Conclusion and Recommendations	145
List of References	147
Appendices	155
A The Optronic Scene System Simulation	156
A.1 Description of OSSIM	156
A.2 OSSIM code structure	157
A.3 Image rendering in OSSIM using the Green, Red and Blue Code	158
A.4 High resolution and sensor images	159
A.5 Random number generators used in OSSIM	160
A.6 OSSIM coordinate systems	160

B The Satellite Body	162
B.1 Description of a satellite body model	162
B.2 The satellite body model in the simulation environment	163
C Test Targets	165
C.1 Test targets used to validate satellite image data	165
C.2 Test target models used in the simulation environment	168
D Derivation of the Atmospheric Aerosol Point Spread Function	173
D.1 Aerosol scattering modulation transfer function	173
D.2 Aerosol scattering point spread function	174

List of Figures

1.1	Charged coupled device array configuration.	10
1.2	Flux against reflectance plot.	12
1.3	Plot of the OSPA calculated digital number output.	13
1.4	Plot of the rotating scan mirror yaw attitude and sensor output image. . .	14
1.5	Plot of the oscillating scan mirror yaw attitude and sensor output image. .	14
1.6	The full circular descend retrograde orbit of the WorldView-2 satellite. . .	16
2.1	Passive satellite remote sensing image chain.	19
2.2	Atmospheric transmittance as a function of wavelength.	22
2.3	The vertical structure of the atmosphere.	24
2.4	Interaction between incident optical flux and the surface of an object. . . .	25
2.5	Comparison of spectral reflectance curves of grass, sand and seawater. . . .	26
2.6	Diffuse, specular and mixed reflectance distribution of objects.	27
2.7	Image acquisition process of an across-track scanning electro-optical sensor.	29
2.8	Image acquisition process of an along-track scanning electro-optical sensor.	29
2.9	Different sensor scanning configurations used to image a scene.	30
2.10	Possible configurations of the charge coupled device lines.	31
2.11	Time delay integration (TDI) process using a three stage TDI detector. . .	32
2.12	Orbits for earth observation remote sensing satellites.	32
3.1	Functional diagram for OSPA remote sensing satellite image capture. . . .	37
4.1	Illustration of the OSSIM signature rendering equation.	44
4.2	The in-field-of-view and out-of-field-of-view adjacent pixel radiance. . . .	47
5.1	Illustration of the angles of a triangle on a sphere.	53
5.2	MODTRAN atmospheric transmittance for two different visibilities.	55
5.3	MODTRAN atmospheric optical depth for two different visibilities.	56
5.4	Atmospheric aerosol modulation transfer function at 776 nm.	58
5.5	Atmospheric aerosol point spread function at 776 nm.	58
6.1	WorldView-2 satellite sensor spectral band definitions.	62

6.2	Spectral reflectance for the FSS99-500 mirror coating.	63
6.3	Sensor optics and detector spatial blurring point spread function.	64
6.4	Charged coupled device array configuration.	65
6.5	WorldView-2 satellite sensor focal plane array layout.	67
8.1	Illustration of the Keplerian orbital elements on an elliptical orbit	80
9.1	Staggered line image correction process.	92
9.2	Alternating scan line image correction process.	94
11.1	Test targets used to evaluate OSPA.	101
11.2	Panchromatic band output image of a Siemens star.	104
11.3	Siemens star calculated modulation transfer function.	105
11.4	Sensor output for a staggered detector focal plane layout.	110
11.5	Sensor output images for a shifted detector array configuration.	112
11.6	Digital numbers produced by shifted detectors scanning along-track.	113
11.7	Plot of the OSPA simulation signal and noise output.	116
11.8	Image scanned across-track using a rotating scan mirror.	120
11.9	Rotating scan mirror attitude against simulation time.	121
11.10	Satellite trajectory coordinates while using a rotating scan mirror.	122
11.11	Image scanned across-track using an oscillating scan mirror.	125
11.12	Oscillating scan mirror attitude against simulation time.	125
11.13	Satellite trajectory coordinates while using an oscillating scan mirror.	126
11.14	Test target scanned using different orbit directions and earth rotation.	129
11.15	Digital numbers produced for different orbit directions and earth rotation.	130
11.16	The full circular descending retrograde orbit of the WorldView-2 satellite.	132
11.17	WorldView-2 satellite attitude for a full circular descending retrograde orbit.	133
11.18	WorldView-2 satellite orbit trajectory between two coordinates.	136
11.19	WorldView-2 satellite attitude for a trajectory between two coordinates.	137
11.20	Along-track staggered detector array layout image correction.	140
11.21	Digital number values for the staggered detector array image correction.	141
11.22	Sensor output image processed for oscillating scan mirror distortion.	143
A.1	OSSIM code structure.	157
A.2	Roles of Green Code, Red Code and Blue Code in image rendering.	158
A.3	High resolution and sensor images used in the image rendering process.	159
A.4	Satellite orientation in the simulation environment.	161
B.1	Primary components of the WorldView-2 satellite architecture	163
B.2	Satellite body model as seen in the simulation environment.	164

C.1	Transportable test targets found at the Sjökkulla test site.	166
C.2	Test targets found at the Stennis Space Center.	166
C.3	Paardefontein calibration and validation test target.	167
C.4	Tarpaulin and painted edge targets and a greyscale bar target model. . . .	168
C.5	Radial edge Siemens star target model.	169
C.6	Paardefontein calibration and validation site model.	170
C.7	Rectangular line target.	170
C.8	Triangular target model.	171
C.9	Triangular target on the surface of a spherical target model.	172

List of Tables

1.1	Description of variables used in the signature rendering equation	8
2.1	Electromagnetic spectral regions used by satellite remote sensing systems. .	21
4.1	Description of variables used in the OSSIM signature rendering equation .	43
6.1	WorldView-2 calculated noise equivalent reflectance values.	70
6.2	List of flux offset and gain factors and detector responsivity values.	73
11.1	In-FOV adjacency effect constant and peak point spread function values. .	105
11.2	In-FOV adjacency effect for a 3.5% reflective target.	106
11.3	In-FOV adjacency effect for a 55% reflective target.	107
11.4	Calculated staggered detector focal plane layout pixel values.	109
11.5	Calculated shifted detector focal plane layout pixel values.	113
11.6	Calculated signal to noise ratio and noise equivalent reflectance values. . .	115
11.7	Calculated digital numbers for a range of target reflectance values.	118
11.8	Pixel distortion present in a processed image.	140

Nomenclature

Constants

A_{Sun}	=	$1.523 \times 10^{12} \text{ km}^2$
d_{Sun}	=	$149.6 \times 10^6 \text{ km}$
g	=	9.81 m/s^2
r_{EARTH}	=	6378.1 km
π	=	$3.141\ 592\ 654$

Variables

\vec{A}^L	Satellite current position Local-Level coordinate vector	[unitless]
$A_a(h, \lambda)$	Atmospheric attenuation absorption coefficient	$[\text{m}^{-1}]$
$A, C, K, L,$ $M, k, l, m,$ u, v, x	Free variables	[unitless]
$A_x^L, A_y^L, A_z^L,$ $B_x^L, B_y^L, B_z^L,$ C_x^L, C_y^L, C_z^L	Local-Level coordinate vector components	[unitless]
\vec{B}^L	Satellite next position Local-Level coordinate vector	[unitless]
\vec{C}^L	Local-Level coordinate vector cross product	[unitless]
$C^G(\varphi, \lambda)$	Earth geographic coordinate	[degree]
C_{const}	Scattering point spread function constant	[unitless]
C_{peak}	Scattering point spread function peak	[unitless]
C_{Φ_p}	Flux offset	[W]
D	Aperture diameter	[m]
D_g	Ground distance increment in the flight direction	[m]
G_{gl}	Greylevel amplification gain factor	[gl/A]
G_{Φ_p}	Flux gain factor	[unitless]
H	Satellite altitude	[m]
I_{inFovAdj}	In-field-of-view adjacency effect irradiance component	$[\text{W}/\text{m}^2]$

$J_1(x)$	First order Bessel function	[unitless]
L_1	Thermal self-emitted atmospheric path radiance	[W/(m ² · sr)]
L_2	Solar scattered atmospheric path radiance	[W/(m ² · sr)]
L_3	Reflected sunlight radiance	[W/(m ² · sr)]
L_4	Self-emitted thermal radiance	[W/(m ² · sr)]
L_5	Solar scattered reflected sky radiance	[W/(m ² · sr)]
L_6	Thermal self-emitted reflected sky radiance	[W/(m ² · sr)]
L_7	Reflected ambient environment radiance	[W/(m ² · sr)]
L_8	Thermal self-emitted ambient environment radiance	[W/(m ² · sr)]
L_9	Solar scattered background-target path radiance	[W/(m ² · sr)]
L_{10}	Thermal self-emitted background-target path radiance	[W/(m ² · sr)]
L_{11}	Reflected sunlight transmitted background radiance	[W/(m ² · sr)]
L_{12}	Thermal self-emitted transmitted background radiance	[W/(m ² · sr)]
L_{13}	Out-field-of-view adjacent reflected solar radiance	[W/(m ² · sr)]
L_{14}	Out-field-of-view adjacent self-emitted thermal radiance	[W/(m ² · sr)]
L_{15}	In-field-of-view adjacent reflected solar radiance	[W/(m ² · sr)]
L_{16}	In-field-of-view adjacent self-emitted thermal radiance	[W/(m ² · sr)]
$L_{inFovAdj}$	In-field-of-view adjacency effect radiance component	[W/(m ² · sr)]
$L_{outFovAdj}$	Out-field-of-view adjacency effect radiance component	[W/(m ² · sr)]
$L_{path\lambda}$	Spectral atmospheric path radiance	[W/(m ² · sr)]
L_{Pi}	Radiance of the i th pixel in the scene	[W/(m ² · sr)]
$L_{Ptarget}$	Radiance of the target pixel	[W/(m ² · sr)]
$L_{sky\lambda}$	Spectral sky radiance	[W/(m ² · sr)]
$L_{\Delta\lambda}$	At-sensor spectral radiance in the wavelength range	[W/(m ² · sr)]
$MTF_a(f_r)$	Atmospheric aerosol modulation transfer function	[unitless]
N_{mirror}	Number of reflective mirrors in the telescope	[unitless]
N_{pixel}	Number of pixels in a scene	[unitless]
P_{gl}	Pixel greylevel	[gl]
P_{noise}	Pixel noise	[gl]
P_{NUgain}	Detector pixel nonuniformity gain	[gl]
$P_{NUoffset}$	Detector nonuniformity offset	[gl]
$PSF_a(\omega_r)$	Atmospheric aerosol point spread function	[unitless]
$PSF_{aNoScatter}(\omega_r)$	Unscattered radiation aerosol point spread function	[unitless]

$PSF_{\text{aScatter}}(\omega_r)$	Scattered radiation aerosol point spread function	[unitless]
PSF_{detector}	Detector point spread function	[unitless]
PSF_{optics}	Optic point spread function	[unitless]
R_{gl}	Detector responsivity in terms of greylevels	[gl/W]
$\mathbf{R}_y^E, \mathbf{R}_z^E$	Earth-Centered Earth-Fixed axes rotation matrix	[unitless]
$\mathbf{R}_x^G, \mathbf{R}_z^G$	Earth geographic axes rotation matrix	[unitless]
\mathbf{R}_{yz}^E	Earth-Centered Earth-Fixed to North-East-Down coordinates conversion rotation matrix	[unitless]
\mathbf{R}_{zx}^G	Orbit orientation rotation matrix	[unitless]
R_λ	Detector responsivity	[A/W]
$S_a(h, \lambda)$	Atmospheric attenuation scattering coefficient	[m ⁻¹]
S_λ	Sensor spectral response	[unitless]
T_a	Ambient environment temperature	[K]
T_{adjin}	In-field-of-view adjacent object temperature	[K]
T_{adjout}	Out-field-of-view adjacent object temperature	[K]
T_b	Background temperature	[K]
T_o	Object temperature	[K]
T_s	Sun temperature	[K]
a	Orbit semi-major axis	[m]
b	Quantisation bin width	[A or V]
$dM/d\rho$	NEM to NE $\Delta\rho$ scaling factor	[W/m ²]
d_g	Focal plane distance increment in the flight direction	[m]
d_{st}	Detector sub-array focal plane stagger distance	[m]
d_x	Distance in the focal plane in the x direction	[m]
d_y	Distance in the focal plane in the y direction	[m]
e	Orbit eccentricity	[unitless]
$f_o(\theta_s, \theta_v, \varphi_s, \varphi_v)$	Bidirectional reflectance distribution function	[sr ⁻¹]
f'	Imaging system focal length	[m]
f	Spatial frequency in integration limits	[cycles/m]
f_c	Spatial frequency cutoff in terms of meters	[cycles/m]
f_c	Spatial frequency cutoff in terms of radians	[cycles/rad]
f_r	Spatial frequency	[cycles/m]
h_t	Target altitude above the ground	[m]
i	Orbit inclination angle	[degree]

i_{\perp}	Angle perpendicular to the track	[rad]
i_{AB}	Inclination angle between coordinate vectors	[rad]
i_{tilt}	X-axis rotation matrix angle	[rad]
p	Pixel size in the focal plane	[m]
p_x	Pixel x dimension size in the focal plane	[m]
p_y	Pixel y dimension size in the focal plane	[m]
r	Radial distance in the focal plane	[m]
r_c^L	Local-Level coordinate cross product radial distance	[unitless]
r_a	Scattering particle radius	[m]
r_{NED}^L	North-East-Down coordinates radial distance	[unitless]
s	Arc length	[m]
t	Time	[min]
v	Satellite platform velocity	[m/s]
x^E, y^E, z^E	Earth-Centered Earth-Fixed coordinates	[unitless]
$x_{\text{ref}}^E, y_{\text{ref}}^E, z_{\text{ref}}^E$	Reference coordinates	[unitless]
x^G, y^G, z^G	Earth geographic coordinates	[unitless]
$x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L$	Local-Level North-East-Down coordinates	[unitless]
ΔC_{end}	Number of image samples in one across-track scan	[unitless]
Δt_s	Integration time	[s]
$\Delta \epsilon_o$	Object spatial texture variation in emissivity	[unitless]
$\Delta \theta_s$	Across-track sweeping angle increment	[rad]
$\Delta \rho_o$	Object spatial texture variation in reflectivity	[unitless]
$\Delta \tau_o$	Object spatial texture variation in transmittance	[unitless]
$\Phi_{\text{absorbed}}(\lambda)$	Absorbed flux	[W]
$\Phi_{\text{incident}}(\lambda)$	Total incident flux	[W]
Φ_p	Pixel flux	[W]
$\Phi_{\text{reflected}}(\lambda)$	Reflected flux	[W]
$\Phi_{\text{transmitted}}(\lambda)$	Transmitted flux	[W]
Ω	Orbit right ascension of the ascending node angle	[degree]
Ω_{rot}	Z-axis rotation matrix angle	[rad]
$\alpha(\lambda)$	Fraction of absorbed flux	[unitless]
$\beta_{\text{ext}}(h, \lambda)$	Extinction coefficient	[unitless]
ϵ	Obscuration area ratio	[unitless]

$\epsilon_{a\lambda}$	Ambient environment spectral emissivity	[unitless]
$\epsilon_{adjin\lambda}$	In-field-of-view adjacent object spectral emissivity . . .	[unitless]
$\epsilon_{adjout\lambda}$	Out-field-of-view adjacent object spectral emissivity . .	[unitless]
$\epsilon_{b\lambda}$	Background spectral emissivity	[unitless]
$\epsilon_{o\lambda}(\theta_v, \varphi_v)$	Object surface directional spectral emissivity	[unitless]
$\epsilon_{s\lambda}$	Spectral emissivity of the surface of the sun	[unitless]
θ	Viewing angle relative to a vertical path	[rad]
θ_a	Vertical to surface normal angle	[rad]
θ_s	Incident solar irradiance vector zenith angle	[rad]
θ_{sw}	Swath width field of view	[rad]
θ_v	Viewing direction vector zenith angle	[rad]
λ	Imaging wavelength	[m]
λ	Longitude	[rad]
λ_{ref}	Reference longitude	[rad]
μ	Gaussian curve mean	[rad]
$\rho(\lambda)$	Fraction of reflected flux	[unitless]
ρ_{mirror}	Reflectance of a single mirror	[unitless]
$\rho_{o\lambda}$	Object surface diffuse spectral reflectance	[unitless]
$\rho_{totalMirror}$	Total reflectance of the mirror optics	[unitless]
σ	Gaussian curve standard deviation	[rad]
$\sigma_{od}(h, \lambda)$	Optical depth along the viewing angle	[unitless]
$\tau(\lambda)$	Fraction of transmitted flux	[unitless]
$\tau_{a\lambda}$	Object-sensor path spectral transmittance	[unitless]
$\tau_{ao\lambda}$	Ambient environment-object path spectral transmittance	[unitless]
$\tau_{bo\lambda}$	Background-object path spectral transmittance	[unitless]
$\tau_{o\lambda}$	Object spectral transmittance	[unitless]
$\tau_{so\lambda}$	Sun-object path spectral transmittance	[unitless]
ν	Orbit true anomaly angle	[degree]
$\phi_{roll}, \theta_{pitch}, \psi_{yaw}$	Satellite attitude angles	[rad]
φ	Latitude	[rad]
φ_{ref}	Reference latitude	[rad]
φ_s	Incident solar irradiance vector azimuth angle	[rad]
φ_v	Viewing direction vector azimuth angle	[rad]

ψ_s	Solar radiance to object surface scaling factor	[unitless]
ω	Orbit argument of perigee angle	[degree]
ω_c	Angular frequency cutoff	[rad]
ω_d	Detector solid angle field of view	[sr]
$\omega_{\text{pixelPitch}}$	Angular frequency for a single pixel	[rad]
ω_r	Angular frequency	[rad]

Abbreviations

2D	Two-dimensional
3D	Three-dimensional
6SV	Second Simulation of a Satellite Signal in the Solar Spectrum Vector code
ADC	Analogue-to-Digital Converter
BRDF	Bidirectional Reflectance Distribution Function
CCD	Charge Coupled Device
CMOS	Complementary Metal-oxide Semiconductor
CSIR	South African Council for Scientific and Industrial Research
DIRSIG	Digital Imaging and Remote Sensing Image Generation
DN	Digital Number
DSA	Detector Sub-array
ECEF	Earth-Centered Earth-Fixed
EM	Electromagnetic
FWHM	Full Width at Half Maximum
FOV	Field of View
FU	Functional Unit
GEO	Geo-synchronous Earth Orbit
GIFOV	Ground-projected Instantaneous Field of View
GSD	Ground Sample Distance
IFOV	Instantaneous Field of View
LEO	Low Earth Orbit
LWIR	Longwave Infrared
MODTRAN	Moderate Resolution Atmospheric Transmission
MTF	Modulation Transfer Function
MWIR	Midwave Infrared

NE $\Delta\rho$	Noise Equivalent Reflectance Difference
NED	North-East-Down
NEE	Noise Equivalent Irradiance
NEM	Noise Equivalent Exitance
NIR	Near Infrared
OSPA	OSSIM for space applications
OSSIM	Optronics System Simulator
PICASSO	Parameterized Image Chain Analysis and Simulation Software
PSF	Point Spread Function
RAAN	Right Ascension of the Ascending Node
RTC	Radiative Transfer Code
SENSOR	Software Environment for the Simulation of Optical Remote sensing systems
SIMIS	Simulator for Imaging Systems
SNR	Signal-to-Noise Ratio
SWIR	Shortwave Infrared
TDI	Time Delay Integration
TMA	Three-Mirror Anastigmatic
TOA	Top-of-Atmosphere
U.S.	United States
XML	Extensible Mark-up Language

Chapter 1

Introductory Summary

1.1 Study topic and motivation

Remote sensing is the process of collecting data that describes the properties of an object of interest without coming into physical contact with that object [1, p. 1], [2, p. 2], [3, p. 1], [4, p. 2.1]. The remote sensing process follows the characteristic process of data acquisition, transmission, reception, processing and analysis, and implementation [3, pp. 2-3], [4, p. 2.1-2.2], [5]. Satellite sensors are used for remote sensing of the earth in order to provide a repetitive and consistent view of a scene. The data collected by these earth observation sensors can be used to monitor both short and long-term changes of the land, water and atmosphere, as well as the impact of human activities [1, p. 2], [2, p. 2].

The quality of a satellite remote sensing system output is affected by any degradation effects or errors introduced at any point in the remote sensing process [1, p. 557]. For instance, the sensor components can degrade the radiometric quality of the image [2, p. 110]. The geometric characteristics of the image can be affected by the sensor platform orbit and attitude, the sensor scanner and the rotation and shape of the earth [2, p. 110]. It is, therefore, important when designing a satellite remote sensing system that the processes are understood and the system requirements are defined [5]. This ensures that the satellite remote sensing system produces the expected results. It is also desirable to reduce the dependency on extensive field tests when developing, testing and calibrating earth observation sensors. A favourable method for aiding the design and testing of a satellite remote sensing system is to make use of a simulation environment to simulate the proposed remote sensing instrumentation and the environment in which it will operate.

The aim of this master's study was to develop a software model of the data acquisition process related to satellite remote sensing for the purpose of image generation. The software model could then be used as a simulation tool to design, optimise and test real world satellite sensors. The simulation tool could be used by satellite sensor design teams to determine system requirements, solve design problems and reduce risks, costs and

development time [5], [6]. The simulation tool would allow a developer to see how the satellite sensor, environment and target would interact with one another for a number of different scenarios. The tool could also be used by image scientists who process and analyse satellite data. It could then be used to develop and evaluate image analysis and processing algorithms for different sensors, as well as to generate sets of training material that cover a wide range of user defined conditions and scenes [1, p. 641], [5], [7, p. 39].

To develop the software model in this study, the data acquisition processes for a passive satellite sensor was investigated and analysed, with the focus lying on the target-atmosphere-sensor relationship. To model the relationship accurately, the target properties, atmospheric effects, sensor architecture and satellite motion were considered. To complete this study, the model was implemented with the aid of the existing simulation software, Optronics System Simulator (OSSIM).

1.2 Study objectives

The objective of this study was to develop a software model of the data acquisition process, specifically the target-atmosphere-sensor relationship, used by a passive satellite remote sensing system. The following study objectives were set for the software model:

- The software model had to serve as a framework which could be further developed to improve on the existing functionality and to include user specific models.
- The radiometry relating to the satellite image acquisition process had to be studied. This process had to be modelled in terms of a signature rendering equation that predicted the at-sensor radiance for a satellite sensor. The equation had to include terms to describe the different paths followed by radiance emitted by the sun or objects. The equation had to take into account the effects on the collected radiance of the atmosphere and sensor response.
- A generic satellite sensor model needed to be implemented that would serve as a framework to model a real world passive satellite sensor. The sensor model had to take into account the sensor response, detector noise and dwell time and include a configurable detector focal plane array layout. The configurable detector array model had to allow the detectors to be staggered and shifted in the focal plane. The sensor model also had to include an analogue-to-digital converter (ADC) functionality to convert the sensor output irradiance into a set of digital number (DN) values.
- The software model had to take into account the method used by the sensor to scan the scene. The model had to simulate the scanning motion of an along-track scanner, as well as the motion of the rotating and oscillating mirrors used in across-track scanners.

- The model had to provide the capability to specify and calculate a satellite orbit trajectory path, excluding the effects of external forces, such as atmospheric drag and gravity, on the satellite trajectory and attitude. The satellite trajectory had to be specified as being either a circular orbit with a specific inclination angle, or a direct flight path between two coordinates.
- The software model had to be able to perform basic image correction to correct for geometric discontinuities that resulted from a staggered detector array layout or an oscillating scan mirror.

1.3 Earth observation using a passive satellite remote sensing system

A satellite remote sensing system describes the end-to-end process followed by an earth observation satellite, starting with the formation of an image from the radiometry in a scene and ending with the final display of the image. This process is also known as the image chain approach and follows the characteristic process of data acquisition, transmission, reception, data processing and analysis, and implementation of the processed data. The focus of this study was on the data acquisition process which is also known as the image capture process [3, pp. 2-3], [4, p. 2.1-2.2], [5]. There are four main elements present in the data acquisition process, which are a source of electromagnetic (EM) radiation, the atmosphere, a target and the satellite platform and sensor [3, pp. 2-3], [4, p. 2.2], [8, pp. 5-6]. Each of these elements affects the quality of the sensor output.

EM radiation is a form of energy that is absorbed, emitted or reflected by objects. EM radiation can be classified as being either passive, originating from a natural source, or active, originating from an artificial source [1, p. 7], [2, p. 10], [4, pp. 2.1 and 2.14]. Remote sensing of EM radiation is described in terms of the spectral range used when collecting data [1, p. 5]. Satellite remote sensing systems primarily collect data in the visible, infrared and microwave regions [1, p. 5], [2, p. 10], [4, p. 2.8]. The different forms of EM radiation provide information associated with that spectral range about the object from which they are reflected or emitted [1, p. 5], [4, p. 2.8].

The atmosphere affects the intensity and spectral composition of the radiance reaching the sensor, by attenuating, amplifying or increasing the propagating radiance [3, p. 9], [9, p. 97]. The primary causes of these effects are atmospheric absorption, scattering and turbulence [1, p. 566], [3, p. 9], [4, p. 2.15], [9, pp. 97 and 108], [10], [11, pp. 443-444]. Atmospheric absorption attenuates the propagating radiation and in so doing changes the apparent spectral signature of the target being imaged [8, p. 14], [10]. Atmospheric scattering changes the direction and spectral distribution of propagating radiance, which causes a change in contrast in the sensor image and also causes the image to appear

hazy or blurred [1, p. 99], [10], [11, p. 443]. Atmospheric turbulence causes deviations in the propagating wavefront, which result in effects such as scintillation, beam steering, image dancing and blurring [1, p. 566], [9, p. 97], [11, p. 444]. Transmittance through the atmosphere is spectrally selective, which results in the transmittance being defined and determined only at specific wavelengths, which are known as the atmospheric transmission windows [9, p. 98]. Atmospheric radiative transfer codes (RTCs), such as Moderate Resolution Atmospheric Transmission (MODTRAN), are used to model the atmosphere by calculating the atmospheric scattering and absorption effects. RTCs are also used when estimating the top-of-atmosphere (TOA) spectral radiance [1, p. 293], [12], [13].

The target being imaged by the satellite sensor is any man-made or natural object of interest. Flux incident on the surface of the target will be absorbed, transmitted and reflected in a manner that is indicative of the object's properties and the physical characteristics of the incident flux [3, pp. 12-13], [4, p. 2.20], [9, p. 27], [14]. The reflectance properties of objects are an important factor when calculating the at-sensor radiance in the visible spectral region [2, p. 13], [3, p. 13].

The sensor is one of the most complex links in the image chain [1, p. 153]. Sensors used in remote sensing systems are classified as being either passive or active sensors [15, p. 9]. Passive sensors measure flux generated by an external source, which is reflected or emitted towards the sensor from the target. Active sensors generate radiation that is beamed down to the target; the sensor then measures the portion of flux reflected back towards the sensor from the target [1, p. 7], [2, p. 10], [15, p. 9]. The sensor converts the collected radiance into an image of the observed scene [2, p. 75]. The sensor system is described by the sensor components, the scanning configuration and the satellite platform and orbit. This study focused on passive imaging electro-optical sensors.

The primary components of an electro-optical sensor system include a scanner, imaging optics, filters or dispersion elements, detectors (for example charge coupled devices (CCD) or Complementary Metal-oxide Semiconductors (CMOS)), electronics and a means of recording data. The sensor electronics include the preamplifier and ADC [1, pp. 167-168], [2, p. 76].

The sensor images a scene by using either an across-track scanner, such as a whiskbroom scanner, or an along-track scanner, such as a pushbroom scanner [2, p. 19], [3, p. 326], [15, p. 20]. An across-track scanner makes use of a mechanical scanning mirror, such as a rotating or oscillating mirror, to scan the scene. An along-track scanner makes use of the platform's motion to provide the scanning action [3, pp. 326 and 332].

Before the detectors collect the optical flux, the flux is separated into different spectral components by means of filters or dispersion elements [2, p. 104], [3, p. 326]. The detectors used to collect the flux form a linear CCD array in the sensor focal plane, where each detector is dedicated to imaging a single row or column of data [3, p. 333], [15, p. 13]. The CCD array is a continuous line that can be segmented, shifted or staggered, or arranged

in a combination of these ways. Each CCD segment is referred to as a detector sub-array (DSA) [1, p. 208], [15, p. 13].

The data recorded by the sensor are converted into a set of DN, using an ADC, before being transmitted to a ground station where the data are processed [1, p. 53], [4, p. 2.31].

Earth observation satellites are commonly placed in low earth orbits (LEO) which are often sun-synchronous, near-polar, descending node orbits [1, p. 49], [2, p. 115], [10], [16, p. 163].

1.4 Overview of the work

The satellite remote sensing system model was developed as a space remote sensing capability in the existing OSSIM environment and is referred to as OSSIM for space applications (OSPA). The OSSIM core library provided the infrastructure and key capabilities upon which this specialist work built. OSSIM was originally designed for short-range, low altitude electro-optical system simulation, such as a staring array sensor mounted on a missile. A description of the OSSIM simulation environment and code structure is provided in Appendix A. The OSSIM staring array sensor model was adapted and additional functionality was added, in order for it to function as a generic satellite sensor model. The OSSIM high resolution image was used to render a sensor image in the OSPA environment. The high resolution image includes the scene content and atmospheric effects. Sensor artefacts and degradation effects were applied to the high resolution image, which was then resampled to produce a lower resolution sensor image [7, pp. 54 and 57]. A description of the high resolution and sensor image is given in Appendix A.4.

The approach to implementing the space remote sensing system model was driven by a functional analysis as discussed in section 3.2. A functional diagram (Figure 3.1) is used to illustrate the dependency of OSPA on OSSIM and how the main steps in the image capturing process are connected and implemented in OSPA. Each block in the diagram is referred to as a functional unit (FU) and represents a capability in the OSSIM and OSPA environment. The OSSIM and OSPA environments were used to set up the simulation scenario. This included setting up the scene to be imaged, the atmosphere, the sensor and scanner and calculating the satellite orbit trajectory and attitude (yaw, pitch and roll). The satellite sensor imaged the scene using an along-track or across-track scanner. If an across-track scanner was used, then the OSPA observer gimbal FU was used to adjust the yaw attitude of the scan mirror. The OSSIM core libraries and the OSPA signature rendering equation FUs were then used to calculate the at-sensor radiance for the location being imaged by the sensor. The point in the scene being imaged by a detector in the sensor CCD array was determined by the satellite position and attitude, scan mirror attitude and the CCD layout. The generic satellite sensor FU used the collected radiance

to generate a geometrically and radiometrically accurate sensor image of the target. Each pixel in the greyscale sensor image indicates the at-sensor radiance, over a specific spectral band, for the target area mapped to that pixel. The different target materials result in different at-sensor radiance values, which is seen as different shades of grey (DN values) in the sensor image. The sensor image can then be processed, if required, by the image processing FU, to remove geometric discontinuities. OSPA produces a number of data files which include an unprocessed and a processed image file, files which provide information regarding the radiometry of the scene and a trajectory file to describe the satellite position and attitude. The OSPA functionality is discussed in detail throughout this study. To test the satellite remote sensing model, the real world WorldView-2 satellite was modelled using the OSPA environment.

A signature rendering equation models the total at-sensor radiance ($W/(m^2 \cdot sr)$) for a specific spectral band, defined by the sensor response. The signature rendering equation terms account for various factors that contribute to the at-sensor radiance [9, p. 281]. Contributing factors include the atmosphere, the source-target-sensor geometry, the target orientation, the surrounding objects and terrain and the optical properties of the target [1, pp. 112 and 117], [3, p. 24], [6], [17]. A detailed description of the OSSIM signature rendering equation is provided in section 4.2.

The OSSIM equation includes a reflected sunlight term to describe the surface-reflected radiance of a target and a self-emitted thermal radiance term to provide information about the target temperature and emissivity. The OSSIM equation also includes terms for the atmospheric path radiance, reflected sky radiance and reflected ambient environment radiance. The reflected ambient environment radiance term is used to describe the trapping effect. The trapping effect describes the radiance from the surrounding environment that bounces a number of times between the target and atmosphere before being collected by the sensor [1, p. 61], [18, p. 9]. The OSSIM equation also includes a transmitted background radiance component. This term accounts for the effects of semi-transparent targets placed in front of a target in a scene. Semi-transparent objects that may be viewed by a satellite include smoke and thin cirrus clouds. The solar radiance contributing terms are the main contributors to the at-sensor radiance when imaging in the visible and near infrared (NIR) spectral ranges.

The OSSIM equation does not, however, account for the adjacency effect seen in satellite images. The adjacency effect occurs when the flux from objects adjacent to the target is scattered by atmospheric aerosol particles into the sensor field of view (FOV), towards the target, as upward path radiance [1, p. 59], [2, p. 49], [7, p. 117], [18, p. 8]. This flux mixes together with the target flux, such that the two flux sources cannot be discriminated in the image [19], [20]. The OSPA signature rendering equation was designed by expanding on the OSSIM signature rendering equation to include an adjacency effect term as shown in (1.4.1),

$$\begin{aligned}
 L_{\Delta\lambda} = & \overbrace{\Delta\rho_o \psi_s \cos\theta_s \int_{\lambda_0}^{\lambda_1} \epsilon_{s\lambda} L_{\lambda}(T_s) \tau_{so\lambda} f_o(\theta_s, \theta_v, \varphi_s, \varphi_v) \tau_{a\lambda} S_{\lambda} d\lambda}^{\text{reflected sunlight}} \\
 & + \overbrace{\Delta\epsilon_o \int_{\lambda_0}^{\lambda_1} \epsilon_{o\lambda}(\theta_v, \varphi_v) L_{\lambda}(T_o) \tau_{a\lambda} S_{\lambda} d\lambda}^{\text{self-emitted thermal radiance}} + \overbrace{\int_{\lambda_0}^{\lambda_1} L_{\text{path}\lambda} S_{\lambda} d\lambda}^{\text{atmospheric path radiance}} \\
 & + \overbrace{\Delta\rho_o \cos\theta_a \int_{\lambda_0}^{\lambda_1} \int_{\text{sky}} L_{\text{sky}\lambda} \rho_{o\lambda} \tau_{a\lambda} S_{\lambda} d\Omega d\lambda}^{\text{diffuse reflected sky}} \\
 & + \overbrace{\Delta\tau_o \int_{\lambda_0}^{\lambda_1} \epsilon_{b\lambda} L_{\lambda}(T_b) \tau_{bo\lambda} \tau_{o\lambda} \tau_{a\lambda} S_{\lambda} d\lambda}^{\text{transmitted background}} \\
 & + \overbrace{\Delta\rho_o \int_{\lambda_0}^{\lambda_1} \int_{\text{amb}} \epsilon_{a\lambda} L_{\lambda}(T_a) \tau_{ao\lambda} \rho_{o\lambda} \tau_{a\lambda} S_{\lambda} d\Omega d\lambda}^{\text{diffuse reflected ambient environment}} \\
 & + \overbrace{\int_{\lambda_0}^{\lambda_1} L_{\text{inFovAdj}\lambda} S_{\lambda} d\lambda}^{\text{in-field-of-view adjacent radiance}} . \tag{1.4.1}
 \end{aligned}$$

Table 1.1 provides a description of each variable used in (1.4.1). Equation (1.4.1) is not mathematically rigorous and serves only to define the various contributors to the at-sensor radiance. A detailed description of (1.4.1) is provided in section 4.3.

For this study, the adjacency effect due only to areas mapped to pixels inside and near the sensor FOV are considered. The OSPA equation adjacency effect term is therefore referred to as the ‘in-FOV’ adjacency term.

The in-FOV adjacency effect can be modelled by an atmospheric aerosol scattering point spread function (PSF), which is convolved with the high resolution image. The atmospheric aerosol PSF is calculated by taking the inverse Fourier transform of the atmospheric aerosol scattering modulation transfer function (MTF), resulting in,

$$PSF_a(\omega_r) = \begin{cases} \sqrt{\frac{\pi}{\sigma_{od}(h, \lambda) \omega_c^2}} \exp\left(-\frac{\pi^2 \omega_r^2}{\sigma_{od}(h, \lambda) \omega_c^2}\right), & \text{if } |\omega_r| > \frac{\omega_{\text{pixelPitch}}}{2} \\ \exp[-\sigma_{od}(h, \lambda)] \delta(\omega_r), & \text{if } |\omega_r| \leq \frac{\omega_{\text{pixelPitch}}}{2}, \end{cases} \tag{1.4.2}$$

where $\sigma_{od}(h, \lambda)$ is the optical depth (unitless), ω_c is the angular cut-off frequency (rad), ω_r is the angular frequency (rad) over which the PSF is calculated and $\omega_{\text{pixelPitch}}$ is the angular frequency (rad) for a single pixel.

The aerosol scattering PSF is discussed in section 5.3.4 and a detailed derivation of the PSF is provided in Appendix D. The aerosol scattering MTF equation, section 5.3.4, is derived in papers by Kopeika [19] and Lie [20]. When plotting the PSF it was found that the PSF is wider than the number of detectors in a sensor CCD array, and the size limits of an image that can be computed within a reasonable simulation turn-around time.

Table 1.1: Description of variables used in the signature rendering equation.

Variable	Description	Variable	Description
$L_{\Delta\lambda}$	At-sensor spectral radiance in the wavelength band $\Delta\lambda$	$\Delta\epsilon_o$	Object spatial texture variation in emissivity
$L_{\lambda}(T_o)$	Spectral black body radiance at the object temperature T_o	$\tau_{o\lambda}$	Object spectral transmittance
$L_{\lambda}(T_b)$	Spectral black body radiance at the background temperature T_b	$\tau_{bo\lambda}$	Background-object path spectral atmospheric transmittance
$L_{\lambda}(T_a)$	Spectral black body radiance at the ambient environment temperature T_a	$\tau_{ao\lambda}$	Ambient environment-object path spectral atmospheric transmittance
$L_{\lambda}(T_s)$	Spectral black body radiance at the sun temperature T_s	$\tau_{so\lambda}$	Sun-object path spectral atmospheric transmittance
$L_{\text{path}\lambda}$	Spectral atmospheric path radiance (emitted and scattered radiance)	$\tau_{a\lambda}$	Object-sensor path spectral atmospheric transmittance
$L_{\text{sky}\lambda}$	Spectral sky radiance (emitted and scattered radiance)	$\Delta\tau_o$	Object spatial texture variation in transmittance
$L_{\text{inFovAdj}\lambda}$	Spectral in-field-of-view adjacency effect radiance	θ_a	Angle between the surface normal and the vertical
$f_o(\theta_s, \theta_v, \varphi_s, \varphi_v)$	Object surface bidirectional reflectance distribution function	θ_s	Zenith angle of the incident solar irradiance vector
$\rho_{o\lambda}$	Object surface diffuse spectral reflectance	θ_v	Zenith angle of the viewing direction vector
$\Delta\rho_o$	Object spatial texture variation in reflectivity	φ_s	Azimuth angle of the incident solar irradiance vector
$\epsilon_{o\lambda}(\theta_v, \varphi_v)$	Object surface directional spectral emissivity	φ_v	Azimuth angle of the viewing direction vector
$\epsilon_{s\lambda}$	Spectral emissivity of the surface of the sun	ψ_s	Scaling factor relating solar radiance to the object surface
$\epsilon_{b\lambda}$	Background spectral emissivity	S_{λ}	Sensor spectral response
$\epsilon_{a\lambda}$	Ambient environment spectral emissivity	λ	Wavelength

This makes it impractical to implement the in-FOV adjacency as a convolution-computed PSF. The in-FOV adjacency effect can, however, be approximated with a constant flux value over the sensor FOV. This is possible because the intensity of the spatially wide scattering PSF appears approximately constant over the much smaller scene in the sensor FOV. This results in all pixels surrounding any target pixel having an equal contribution to the in-FOV adjacency effect for that target pixel. The in-FOV adjacency effect term

in (1.4.1) is rewritten in terms of the constant value as,

$$L_{\text{inFovAdj}} = (C_{\text{const}} \sum_{i=0}^{N_{\text{pixel}}} L_i) - (C_{\text{const}} L_{\text{Ptarget}}) + (C_{\text{peak}} L_{\text{Ptarget}}), \quad (1.4.3)$$

where C_{const} is the constant in-FOV adjacency value over the scene and C_{peak} is the PSF peak value at the target pixel. N_{pixel} is the number of pixels in the high resolution image, L_i ($\text{W}/(\text{m}^2 \cdot \text{sr})$) is the i^{th} high resolution image pixel radiance value and L_{Ptarget} ($\text{W}/(\text{m}^2 \cdot \text{sr})$) is the pixel radiance for the target pixel. Equation (1.4.3) was implemented in OSPA as outlined in listing 4.1.

To validate the OSPA in-FOV adjacency effect term, the sensor was set to scan over a Siemens star target and take snap-shot images of a high contrast tarpaulin edge target. The sensor image of the Siemens star is used to determine the effect of the in-FOV adjacency effect on the system MTF. The sensor DN output values for the high contrast target are used to calculate the change in image contrast. This validation is discussed in detail in section 11.2. The test results showed that the in-FOV adjacency effect due to aerosol scattering has a negligible effect on the system MTF. The image of the tarpaulin target, however, shows a change in contrast. The in-FOV adjacency effect changes the image contrast by causing the brighter pixels to appear darker and the darker pixels to appear lighter. The results also show that the brighter pixels are less affected by the adjacency effect than are the darker pixels. These results match the expected results, which confirm that the in-FOV adjacency effect can be successfully modelled as a constant value over the scene.

Each term in the signature rendering equation describes the path followed by the radiance through the atmosphere and therefore each term must take into account the atmospheric transmittance. The atmospheric transmittance is calculated with the aid of the MODTRAN4 RTC. MODTRAN4 was set to model the atmosphere for good visibility conditions above the Pretoria area ($25^{\circ}29'45''\text{S}$, $28^{\circ}22'25''\text{E}$) in March at the estimated WorldView-2 satellite overpass time of 10:30 a.m. The top of the MODTRAN atmosphere is set to be 100 km above the surface of the earth. The MODTRAN4 atmospheric settings are discussed in detail in section 5.3.

Each term in the signature rendering equation is weighted by the sensor spectral response. The sensor spectral response is defined by the sensor spectral band definitions, the detector response and telescope reflective optics. The WorldView-2 spectral bands comprise nine bands, a panchromatic band and eight multispectral bands in the spectral range of $0.45 \mu\text{m}$ to $1.040 \mu\text{m}$ (Figure 6.1) [21]. The reflective optics consist of four mirrors in a Korsch type II three-mirror anastigmatic (TMA) telescope design [22]. The mirrors are coated with FSS99-500 silver metallic coating from Quantum coating (Figure 6.2) [23]. The detector response was modelled by the detector responsivity equation as provided in [9, pp. 253 and 415].

The generic sensor model is the core model of this study and was designed and implemented to allow a user to simulate different sensor configurations. The sensor model includes a configurable CCD array (section 6.3.4), detector noise (section 6.3.6) and non-uniformity (section 6.3.7), telescope optics (section 6.3.2) and an ADC (section 6.3.8). The sensor parameters in each of the sensor spectral bands are handled separately, with potentially different sensor configurations.

The sensor model CCD array size is set for each imaging spectral band, by specifying the detector size and the number of detector element rows and columns. The CCD array can be divided into even and odd DSA groups, which can be staggered along or across the track with no overlapping detectors between DSA groups. The full CCD array for an imaging spectral band can also be shifted along or across the focal plane. Figure 1.1 illustrates the sensor model CCD array configuration.

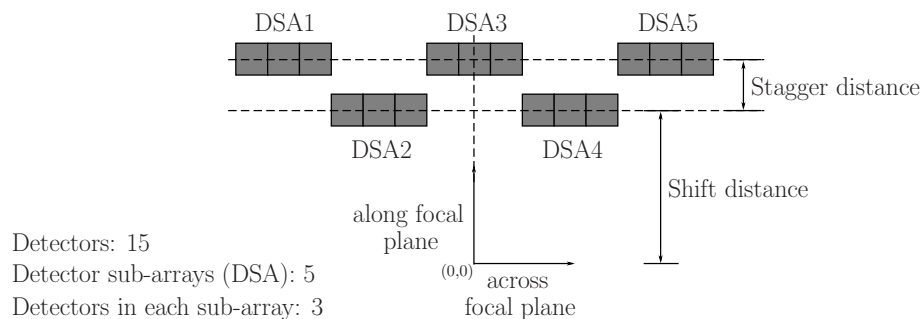


Figure 1.1: Charged coupled device array configuration illustrating the along-focal plane stagger and shift distances.

Listing 6.1 provides the pseudo code used to calculate the focal plane distance by which to adjust each detector.

The stagger and shifted focal plane layout is tested in sections 11.3 and 11.4. These tests model a CCD array focal plane layout similar to that of WorldView-2 (Figure 6.5) with the exception of fewer detectors in a CCD array. The panchromatic band CCD is set to be a single row of 600 detectors that are divided into 50 staggered DSAs. Each multispectral band CCD is set to be a single row of 150 detectors that are divided into 10 staggered DSAs. The CCD arrays are shifted along the focal plane so that the NIR 1 band images the scene first, followed by the rest of the bands in the order of red, green, blue, panchromatic, red edge, yellow, coastal blue and NIR 2. The sensor is set to image a line target. The sensor output images clearly show that the image of the line target appears to change position along the track from one imaging band to the next. This indicates that the bands image the target at different times. The staggered array distortion is also visible in the sensor images where the line target appears to be broken into smaller segments, arranged along two rows in the image. The stagger and shift distances calculated from the

target position and distortion in the images are shown to be correct. The sensor output images for these tests confirm that the OSPA simulation environment is able to model a staggered and shifted detector array configuration correctly.

The detector noise is added to the model by adding a Gaussian distribution random number sequence to the pixel signals [24, p. 194]. The standard deviation of the Gaussian distribution is set as the sensor noise equivalent irradiance (NEE) (W/m^2) and the mean value is set to 0 [9, p. 258]. The sensor NEE value for a Lambertian diffuse reflecting surface is calculated using noise equivalent reflectance $\text{NE}\Delta\rho$ (unitless), noise equivalent exitance (NEM) (W/m) and a scaling factor ($dM/d\rho$) (W/m^2),

$$\text{NEE} = \frac{\omega_d}{\pi} \frac{dM}{d\rho} \text{NE}\Delta\rho \quad (1.4.4)$$

where ω_d (unitless) is the detector solid angle FOV, calculated as $\omega_d \approx (p_x p_y)/(f')^2$ since the pixel IFOV is small [9, pp. 241-242]. p_x and p_y are the detector dimensions (m) and f' is the imaging system focal length (m). The derivation of (1.4.4) is discussed in section 6.3.6.

The sensor noise is tested in section 11.5 by imaging a striped line target. The reflectance of the line target stripes is set such that the sensor output DN signal will have a signal-to-noise ratio (SNR) of 10. This test is used to confirm that the correct noise signal is observed in the output signal for a given $\text{NE}\Delta\rho$ value. If the detector noise is calculated correctly, then the output signal will produce the correct SNR. For instance, the signal produced by the panchromatic band had a value of 26.2265 gl with a noise value of 2.6191 gl. The SNR was then calculated to be 10.0136 which is 0.136% bigger than the expected SNR of 10. The test results showed that the sensor output signals for each WorldView-2 spectral band produced a SNR to within 0.3% of the desired SNR of 10. The results confirm that the sensor noise can be modelled accurately using the $\text{NE}\Delta\rho$.

The detector output electronic signal is processed and undergoes analogue-to-digital conversion before transmission to a ground station [1, p. 167]. The generic sensor model provides the functionality to calculate the DN value, also known as the greylevel value, of a pixel. The DN value is based on the definition of the gain, offset and ADC resolution, as discussed in section 6.3.8. The WorldView-2 satellite sensor has a theoretical ADC resolution of 2048 available greylevels. Most sensors set safety margins at the lower and upper quantisation range limits to prevent the output from saturating on occasionally high or low signal events, noise events, or when doing non-uniformity correction. For this study the full quantisation range was set as being 410 gl to 1638 gl. The pixel greylevel for this quantisation range was calculated using

$$P_{\text{gl}} = \left[(G_{\Phi_p} \Phi_p + C_{\Phi_p}) R_{\text{gl}} + P_{\text{noise}} \right] P_{\text{NUgain}} + O_{\text{gl}} + P_{\text{NUoffset}}, \quad (1.4.5)$$

where P_{gl} is the pixel greylevel value (gl), P_{noise} is the pixel noise (gl), O_{gl} is the sensor lower DN offset (gl) and P_{NUgain} and P_{NUoffset} are the detector non-uniformity gain and

offset (gl) respectively. Φ_p is the pixel flux (W), C_{Φ_p} (W) and G_{Φ_p} (unitless) are the flux offset and gain values respectively and R_{gl} is the detector responsivity in terms of greylevels (gl/W).

The pixel flux being converted by (1.4.5) is calculated from the measured radiance calculated in (1.4.1). The radiance (W/(m² · sr)) is converted into flux (W) by multiplying the radiance by the solid angle of the detector IFOV (sr), the detector fill factor (unitless) and the optics collection area (m²). Before the pixel flux is converted it must be adjusted to account for any offset and gain changes caused by the atmosphere. The flux offset is equivalent to the atmospheric path radiance and the gain is calculated, as discussed in section 6.3.8, as the change observed in the gradient of the ideal flux against reflectance graph when an atmosphere is added. Figure 1.2 illustrates the flux against reflectance curves. The ideal curve with no atmosphere has no offset but when an atmosphere is present an offset is added and the gradient of the curve changes.

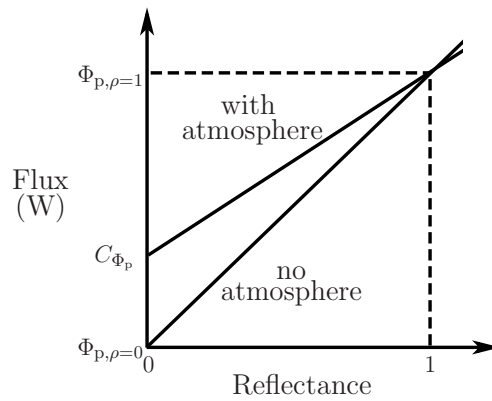


Figure 1.2: Flux against reflectance plot for no atmosphere and with an atmosphere added.

The ADC algorithm requires the greylevel offset and gain values to convert the flux into a DN value. The offset is used to set the lower limit safety margin of the quantisation range. The greylevel gain is expressed by the detector responsivity in terms of greylevel per watt. The gain is used to set the upper limit safety margin of the quantisation range. Listing 6.2 provides the python pseudo code snippet used to calculate the detector responsivity value. The ADC algorithm is calibrated for a specific set of atmospheric conditions.

The ADC algorithm is tested in section 11.6 by imaging a set of targets of known reflectance. A greyscale bar target with reflectances ranging from 5% to 95% was imaged to test that the correct DN value was produced for the corresponding reflectance value. The ADC output calculated for the panchromatic imaging band when imaging the greyscale bar target is plotted in Figure 1.3. The plot shows that the DN values remain within the set quantisation range and remain constant for each image pixel with the same reflectance

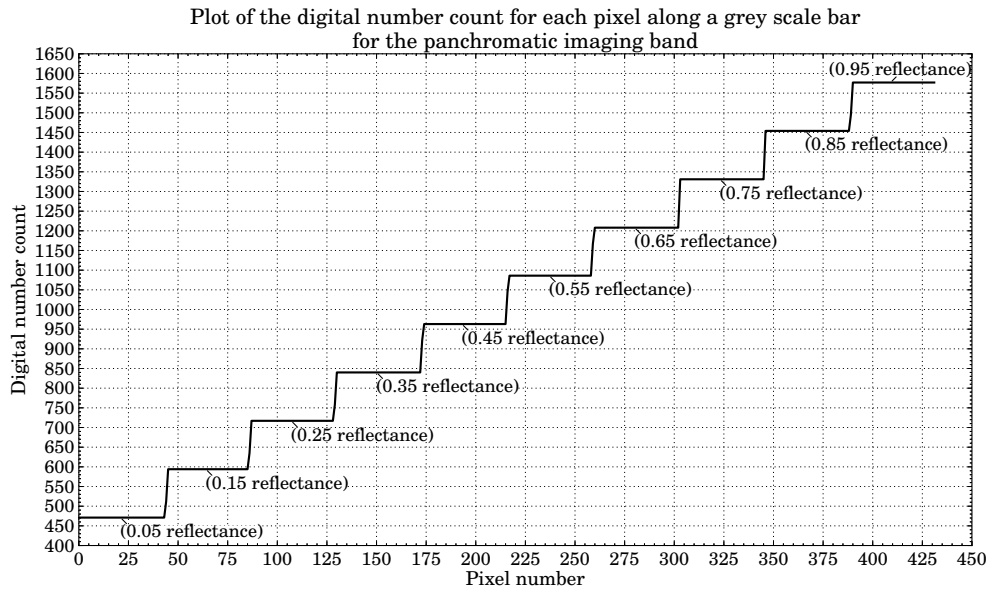


Figure 1.3: Plot of the OSPA simulation digital number output for the WorldView-2 satellite panchromatic band when imaging a greyscale bar target.

value, as seen in the step shape of the graph. Reflective target blocks of 0% and 100% were also imaged to test the upper and lower limits of the quantisation range. The test results showed that the algorithm was able to calculate the correct DN value between the set quantisation range safety margin limits for a reflectance value.

The OSPA environment allows the sensor to scan the scene using either an along-track scanner or an across-track scanner, as discussed in Chapter 7. The scanning motion is simulated by adjusting the satellite attitude and position as a function of time, synchronised with the detector sampling interval.

Along-track scanning is simulated by setting the sensor line-of-sight to point downwards, such that the CCD array lies across the full swath width perpendicular to the along-track direction. This allows a single across-track image line to be sampled each time the sensor moves one ground sample distance (GSD) along the track.

The across-track scanning motion is simulated by setting the sensor line-of-sight to point downwards, such that the CCD array lies along the track perpendicular to the across-track scan direction. The sensor line-of-sight is adjusted in the across-track direction (sideways), to point towards one end of the swath scan. The sensor line-of-sight is then incrementally adjusted across the track at successive sampling intervals. The satellite position is adjusted to move one GSD along the track each time the full swath width has been scanned. A triangular target was scanned using the panchromatic band. The material properties of the left half of the triangle are set to have a reflectance of 95% (appears light grey in the sensor image) and the right half has a reflectance of 45% (appears dark grey in the sensor image).

To simulate a rotating mirror (sawtooth scan), the scan mirror algorithm is set to

calculate the scan mirror attitude such that the swath line is scanned in the same direction across the track for each scan. The rotating across-track scan mirror angle algorithm is validated in section 11.7. Figure 1.4, plots the rotating mirror across track attitude and the sensor output image. The attitude plot has a sawtooth shape indicating that the scan direction is the same for each scan line. This is confirmed by the sensor output image that shows no geometric distortion in the triangular target.

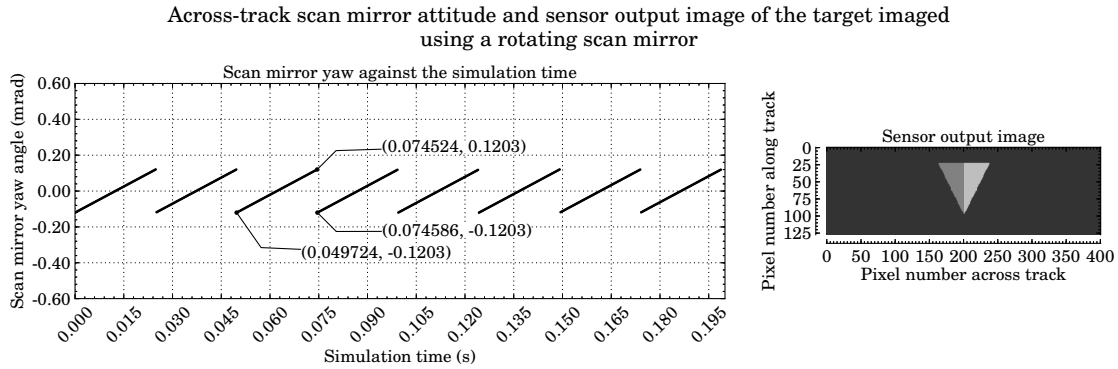


Figure 1.4: Plot of the rotating scan mirror yaw attitude and sensor output image.

To simulate an oscillating mirror (triangular scan), the scan mirror algorithm is set to calculate the scan mirror attitude such that successive swath lines are scanned in opposite directions across the track. The oscillating across-track scan mirror angle algorithm is validated in section 11.8. Figure 1.5, plots the oscillating mirror across track attitude and the sensor output image. The attitude plot has a triangular shape indicating the scan direction is in opposite directions for successive scan lines. This is confirmed by the sensor output image that shows geometric distortion in the triangular target as seen by the pixels corresponding to low reflectance (dark) areas of the target and the pixels corresponding to high reflectance (light) areas of the target which appear on opposite sides of the triangle in alternating scan lines. The displaced pixels show that the scanner started imaging the target from the opposite side with each successive scan line.

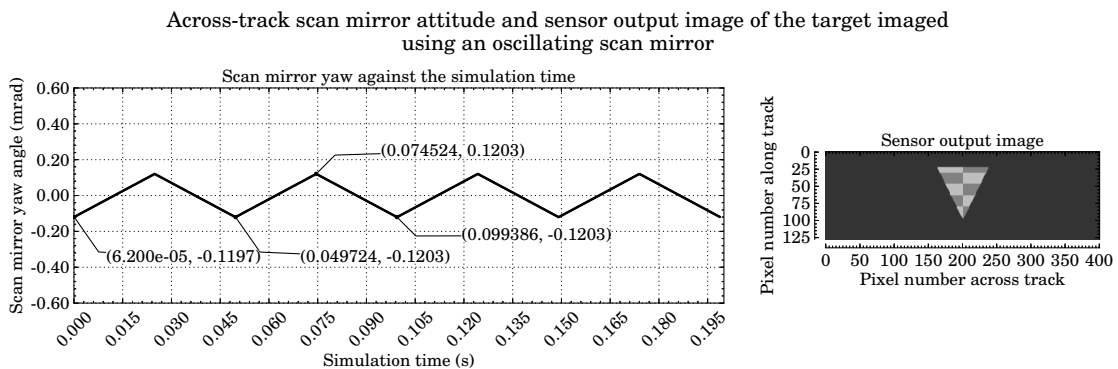


Figure 1.5: Plot of the oscillating scan mirror yaw attitude and sensor output image.

The test results showed that the expected distortion was produced in the sensor images and that the across-track scan mirror attitude and trajectory had been calculated correctly.

The satellite orbit and attitude determine the view the satellite will have of earth [25]. A satellite orbit trajectory model was designed to calculate the satellite position and attitude, for a user-defined satellite orbit. The model provides the functionality to calculate a full orbit trajectory around the earth or a trajectory path between two points on the earth's surface. The model assumes a circular orbit around a perfectly spherical earth, and that no external forces on the satellite are present. The satellite trajectory is calculated for a Local-Level North-East-Down (NED) world coordinate system and a yaw-pitch-roll Euler rotation sequence. The orbit trajectory position and attitude calculation steps are discussed in detail in section 8.3.

A basic orbit trajectory model, section 8.2, was designed to describe a simplified satellite orbit. The full circular orbit trajectory path is described by the orbit altitude, inclination angle and right ascension of the ascending node (RAAN) angle. The trajectory path between two points is described by the orbit altitude and the start and end latitude and longitude coordinates. The orbit trajectory path is calculated relative to a reference coordinate placed at the top of the earth ($90^{\circ}0'0''\text{N}$, $0^{\circ}0'0''\text{E}$). The position coordinates along the trajectory path are evenly spaced and maintain a constant altitude above the surface of a perfectly spherical earth. The orbit direction is described as being ascending when the satellite travels from the southern to northern hemisphere and descending when traveling from the northern to southern hemisphere.

The orbit direction and the rotating earth will distort the sensor output image, as discussed in section 11.9. If an ascending orbit is used to move over a scene, then the north and south directions of the scene in the output image will appear to have been flipped. Similarly, if a descending orbit is used to move over a scene, then the east and west directions of the scene in the output image will appear to have been flipped. A rotating earth will cause a shearing effect in the output image.

The orbit trajectory model was tested to ensure that the satellite followed the desired trajectory path and remained pointing towards the centre of the earth so as to view the scene at nadir. This was tested in section 11.10 for a full circular orbit that matched the WorldView-2 satellite orbit. The model was also tested in section 11.11 for a short trajectory path that lies along the WorldView-2 satellite orbit. The orbit was set to be a descending circular orbit with an inclination angle of 97.8° , a RAAN angle of 180° and an altitude of 770 km. The full orbit trajectory is plotted in Figure 1.6.

The trajectory clearly shows that the calculated satellite ground track is symmetrical and at each marked point in the trajectory the orbit altitude is seen to be 770 km. Furthermore, the orbit passes over the desired RAAN angle and the inclination angle was calculated as being $180.0^{\circ} - 82.2^{\circ} = 97.8^{\circ}$. The results also showed that the satellite

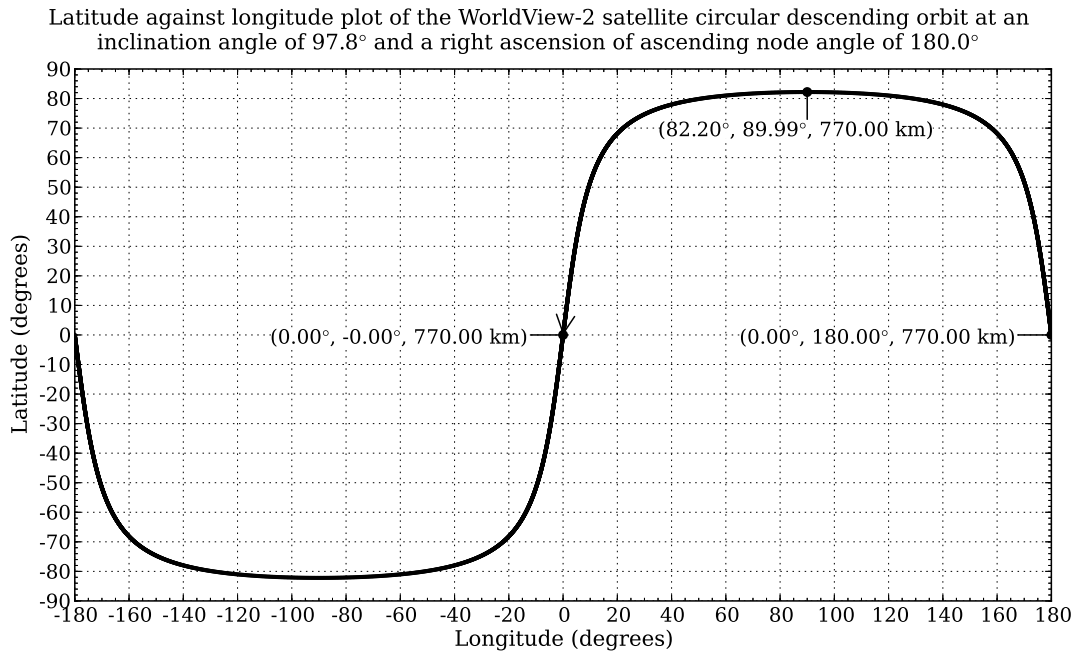


Figure 1.6: The full circular descend retrograde orbit of the WorldView-2 satellite.

attitude remained pointing towards the earth's centre. The test results showed that the orbit trajectory algorithm provided a good model of the satellite orbit trajectory and attitude.

The raw satellite data requires processing before a usable image can be created [5]. The OSPA model includes a basic image pre-processing algorithm to correct for geometric discontinuities in the image caused by a staggered detector array and an oscillating scan mirror. The stagger correction algorithm works by shifting the image pixels along an image column or row until the pixels align with the correct pixels in the adjacent image columns or rows. The algorithm adds interleaving pixels to the image where there is no image data available with which to align the pixels being shifted. The oscillating scan line correction algorithm shifts the pixels across the full image row since the whole scan line is assumed to be reversed. As a result no interleaving pixels are required to be added. The image correction algorithms are discussed in detail in Chapter 9. The image correction algorithms are tested in section 11.12 and section 11.13. The tests show that the algorithm can correct an image for alternating scan lines, and can remove the stagger distortion, provided that the algorithm is set to correct the image for the correct number of pixels.

A detailed discussion on the implementation issues and limitations of the software model is provided in Chapter 10.

The main implementation issues when simulating a satellite remote sensing system are the available hardware memory and the computation time. The available hardware memory restricts the size of the data sets used in the simulation. The computation time

increases with an increase in the size of the data set to be processed. These issues arise when implementing the aerosol scattering PSF and simulating a large detector focal plane array. These issues are addressed by limiting the size of the data sets used or by using an alternative method to simulate the desired effect.

The main limitation in the satellite remote sensing system model is the level of detail with which some features are modelled. These features include the real world scene and targets, the satellite orbit and the exclusion of a time delay integration (TDI) model in the sensor model. These limitations can be addressed in future work.

1.5 Contributions and future work

The main contribution of the work completed in this study was the development of a satellite remote sensing system software model that includes the basic functionality required to simulate the target-atmosphere-sensor relationship for a real world satellite sensor. Of note in this study is the inclusion of the in-FOV adjacency effect in the satellite signature rendering equation. The in-FOV adjacency effect can be modelled by an atmospheric aerosol scattering PSF which is convolved with the high resolution scene image. However, due to the large size of the PSF convolution kernel, this was not possible. It was found that the in-FOV adjacency effect could be approximated with a constant flux value over the sensor FOV as described in sections 4.3, 4.4 and 5.3.4. This approximation was found to be a satisfactory model for the in-FOV adjacency effect, since the model produced the desired effect of a reduction in contrast in the sensor output image. The model contributed to the expansion of the existing OSSIM simulation environment by including a space remote sensing capability. The OSSIM environment proved to provide the necessary infrastructure and key capabilities upon which to develop the satellite remote sensing system software model. The satellite remote sensing system software model allows for further expansion.

Future work can include the expansion of the adjacency effect model to include the out-of-FOV adjacency effect and the inclusion of highly detailed world scene and target models. The OSPA model can be further improved with future work to expand on the sensor, scanner and orbit models. The sensor model can be expanded to include time delay integration and to account for additional noise sources such as quantisation noise and non-uniformity due to read out registers. The scanner model can be expanded to include a scan line corrector. The orbit trajectory model can be expanded to simulate orbital decay due to external forces and to calculate the trajectory of a satellite in an elliptical orbit. Future work also includes the inclusion of more advanced image correction algorithms to correct for atmospheric effects.

Chapter 2

Earth Observation Using Satellite Remote Sensing

2.1 Introduction

Remote sensing is the process of collecting data that describes the properties of an object of interest without coming into physical contact with that object [1, p. 1], [2, p. 2], [3, p. 1], [4, p. 2.1]. Sensors are used to remotely collect data, that are analysed to obtain information about an object, area or phenomenon under investigation [3, p. 1], [4, p. 2.1]. This study focuses on the remote sensing process of an earth observation satellite, which is referred to as the satellite remote sensing system.

Satellite remote sensing systems provide a means of collecting data from a vantage point that is not accessible on the earth. Satellite sensors are designed to view either the earth or the universe around the earth [5]. Satellites used for overhead earth observation, provide a repetitive and consistent view of the earth and allow scenes to be viewed in ways that the human visual system cannot [1, p. 2], [2, p. 2]. Images can be taken in different spectral regions in order to extract different information about a scene. Such information includes the spatial relationships between ground objects and the way in which different objects reflect and emit radiance [2, pp. 7-8], [3, p. 2]. Spaceborne sensors are also used to monitor both short-term and long-term changes of the land, water and atmosphere of the earth, as well as the impact of human activities [1, p. 2], [2, p. 2].

2.2 The satellite remote sensing system

A remote sensing system follows a characteristic process of data acquisition, data processing and displaying of the output [1, p. 1]. The end-to-end process of forming an image, from the radiometry of a scene to the final display of the image, is thought of as a chain of events and is therefore referred to as the image chain approach [1, p. 17], [5].

By following the image chain approach, the five main processes of a satellite remote sensing system are identified [3, pp. 2-3], [4, p. 2.1-2.2], [5].

1. Image capture (data acquisition).
2. Transmission of the recorded data.
3. Reception of the data by a satellite receiving station.
4. Processing, interpreting and analysing the data.
5. Application of the processed data results.

The five main processes, for a passive satellite sensor, are depicted in Figure 2.1.

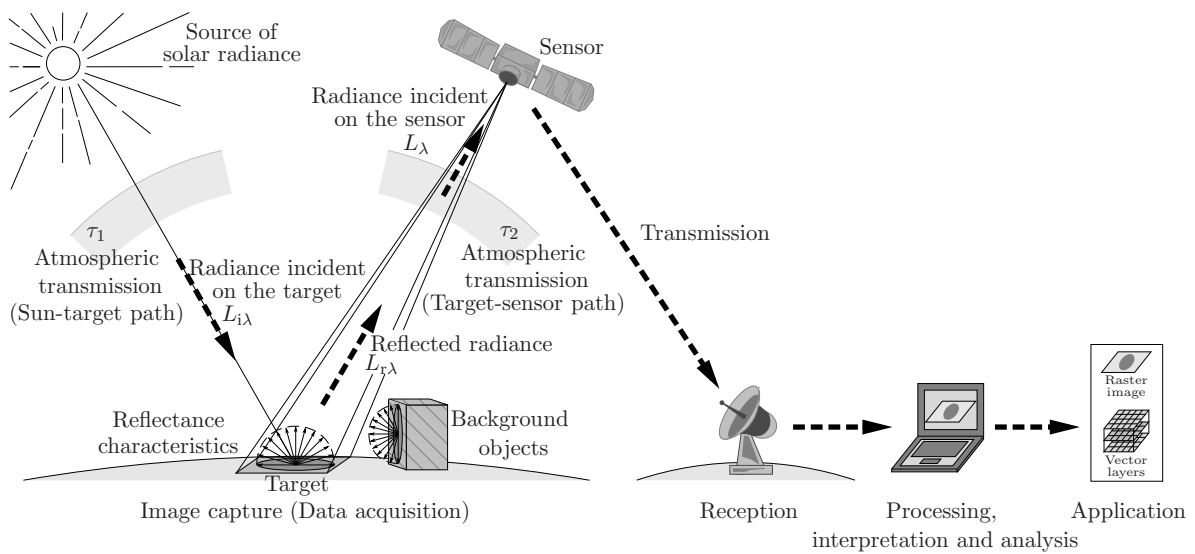


Figure 2.1: Passive satellite remote sensing image chain. The image chain follows the process of data acquisition, transmission, reception, analysis and implementation (Image adapted from [3, p. 3], [4, p. 2.2] and [8, p. 5]).

A satellite remote sensing system requires a source of illumination, such as the sun, or radiance generated and transmitted from the sensor [3, p. 2], [5], [8, p. 7], [26]. The radiance emitted from the illumination source passes through the atmosphere of the earth. The propagating radiance interacts with particles in the atmosphere, which results in phenomena such as absorption and scattering [3, p. 2], [4, p. 2.2], [8, p. 5], [26]. The radiance that reaches the surface of the earth is then reflected off objects, namely the target and ambient environment. The radiance is reflected in a manner that is indicative of the properties of the surface objects. The reflected radiance passes through the atmosphere again and enters the sensor collection aperture. The radiance reaching the sensor is referred to as the at-sensor radiance [3, p. 2], [8, p. 5], [26]. The collected optical signal is processed into a set of digital numbers (DN), which are referred to as the image data set

[4, p. 2.29], [5]. A DN is also referred to as a greylevel or digital level. The image data set is then transmitted to a satellite receiver station [8, p. 6], [26]. At the receiver station the data are processed and interpreted visually or numerically with the aid of a computer [5], [8, p. 6]. The processed data are then supplied to a user for their own use [3, p. 2].

The quality of the system output is affected by any degradation effects or errors introduced at any point in the chain [1, p. 557]. When designing a satellite remote sensing system it is therefore important that the image capture process is understood and that the system requirements are defined [5]. This ensures that the satellite remote sensing system produces the expected results.

The main focus of this study is on the image capture (data acquisition) processes of the satellite remote sensing system.

2.2.1 Image capture (data acquisition)

The data acquisition process comprises four main elements, which are also illustrated in Figure 2.1 [3, pp. 2-3], [4, p. 2.2], [8, pp. 5-6]:

1. Electromagnetic (EM) radiation, which originates from a source.
2. The atmosphere, which interacts with the propagating EM radiation.
3. Earth surface features (the target).
4. The sensor and satellite used to record the at-sensor radiance.

2.2.1.1 Electromagnetic radiation

EM radiation is a form of energy that is absorbed, emitted or reflected by objects and is classified as being either passive or active [4, pp. 2.1 and 2.14]. Passive radiation originates from natural sources such as the sun, sky and earth. Active radiation originates from artificial sources, such as microwave radiation from an electronic device [1, p. 7], [2, p. 10], [4, p. 2.14].

Remote sensing of EM radiation is described in terms of the spectral range used when collecting data [1, p. 5]. The different forms of EM radiation are divided into different regions along the EM spectrum [4, p. 2.8], [27]. These regions include the visible, infrared and microwave regions, which are those primarily used in satellite remote sensing applications [1, p. 5], [2, p. 10], [4, p. 2.8]. These three regions are divided into subregions, which are listed in Table 2.1. The boundary values for the EM regions may differ between sources found in literature.

Each of the different forms of EM radiance provides information associated with that spectral range about the object from which it is reflected or emitted [1, p. 5], [4, p. 2.8]. The information extracted from each of the EM spectral regions can be combined to improve interpretation and analysis of the image data [2, p. 12].

Table 2.1: Electromagnetic spectral regions used by satellite remote sensing systems [2, p. 10], [4, pp. 2.9 and B.1], [9, p. 21], [27]. Note that the literature does not agree on the exact boundaries between different regions.

Electromagnetic spectral region	Spectral region subdivision	Wavelength range (μm)
Visible light	Violet	0.400 – 0.446
	Blue	0.446 – 0.500
	Green	0.500 – 0.578
	Yellow	0.578 – 0.592
	Orange	0.592 – 0.620
	Red	0.620 – 0.700
Infrared	Near Infrared (NIR)	0.700 – 1.100
	Shortwave Infrared (SWIR)	1.100 – 1.350
		1.400 – 1.800
		2.000 – 2.500
	Midwave Infrared (MWIR)	3.000 – 4.000
		4.500 – 5.000
Longwave Infrared (LWIR)	8.000 – 9.500	
	10.000 – 14.000	
Microwave	millimetre	1000 – 10 000
	radar	10 000 – 1 000 000

2.2.1.2 The atmosphere

Transmittance through the atmosphere is spectrally selective, which results in the transmittance being defined and determined only at specific wavelengths [9, p. 98]. The atmosphere appears transparent in wavelength regions of measurable transmittance. These regions are known as atmospheric transmission windows [3, p 10], [9, p. 116], [10]. The atmosphere appears opaque in wavelength regions of negligible transmittance. These regions are known as absorption bands [9, p. 116], [10], [11, pp. 445-446]. Figure 2.2 illustrates the atmospheric transmittance curve as a function of wavelength.

The atmosphere affects the intensity and spectral composition of the radiance reaching the sensor by attenuating, amplifying or increasing the propagating radiance. The primary cause of these effects is absorption and scattering, by particles in the atmosphere, and atmospheric turbulence, which results in random variations in the atmospheric refractive index [1, p. 566], [3, p. 9], [4, p. 2.15], [9, pp. 97 and 108], [10], [11, pp. 443-444].

Atmospheric absorption: Atmospheric absorption attenuates the propagating radiance and converts the radiance into another form of energy, such as thermal energy [1, p. 93], [8, p. 14], [10]. Not only does atmospheric absorption change the apparent spectral

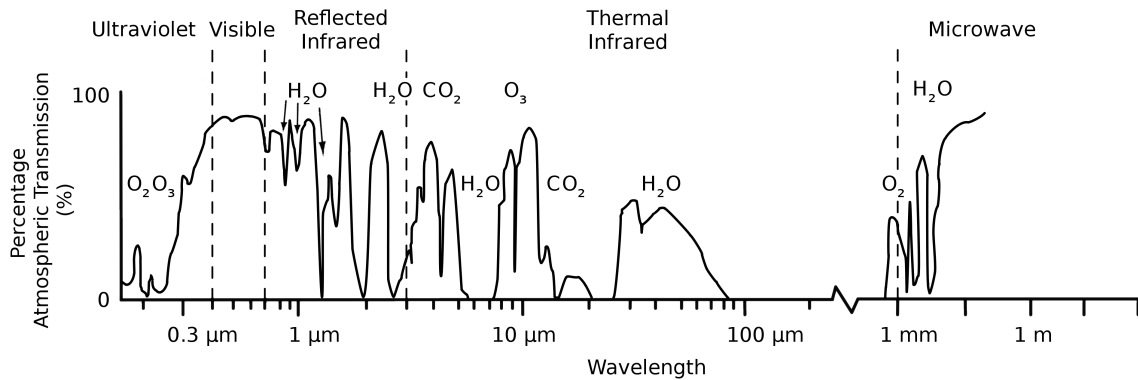


Figure 2.2: Atmospheric transmittance curve with transmission windows and absorption bands as a function of wavelength. The chemical formula is given for the atmospheric gases responsible for the absorption of radiation at a particular wavelength in the spectrum. (Image adapted from [3, p. 11] and [4, p. 2.17]).

signature of the target being imaged, but it also limits the spectral range in which the target can be imaged [8, p. 14], [10].

Atmospheric absorption is wavelength selective [9, p. 111], [11, p. 445]. Strong absorption bands are found in the ultraviolet and infrared regions [11, p. 445]. Hardly any atmospheric absorption is found in the visible spectrum and microwave region [8, pp. 14-15], [11, p. 445]. Ozone, carbon dioxide and water vapour are the atmospheric gases that are the main absorbers of radiation [3, p. 10], [4, p. 2.17].

Atmospheric scattering: Atmospheric scattering changes the direction and spectral distribution of propagating radiance [1, p. 99].

Atmospheric scattering has a number of different effects on the sensor image. For example, the image appears hazy when solar radiation is scattered towards the sensor without first reaching the target [10]. The contrast in the image decreases if flux from areas that map to neighbouring pixels inside and outside the sensor field of view (FOV) is scattered into the FOV of the target pixel [10], [11, p. 443]. The target in the image appears blurred when the propagating radiance is scattered by only small angles such that the radiance from the same point in the scene has multiple angles of arrival at the sensor [10], [11, p. 443]. The scene radiance is attenuated if the radiance is scattered by large angles to such an extent that it is no longer received by the sensor [11, p. 443].

The type of scattering that takes place depends on the ratio of the atmospheric particle radius to the radiance wavelength [11, p.446]. The types of atmospheric scattering include Rayleigh, Mie and non-selective scattering [1, p. 99], [3, p. 10], [4, p. 2.15].

Rayleigh scattering occurs when the wavelength of the propagating radiance is larger than the physical size of the atmospheric molecule with which it is interacting [1, p. 99], [3, p. 9], [4, p. 2.15], [9, p. 115], [10], [11, p. 446]. Short wavelength radiance, below 3 μm,

will be scattered more than long wavelength radiance. Rayleigh scattering is the primary cause of haze in an image and of the blue sky observed on the earth [3, p. 10], [9, p. 115].

Mie scattering occurs when the wavelength of the propagating radiance is equal to the physical size of the atmospheric molecule with which it is interacting [1, p. 99], [3, p. 10], [4, p. 2.16]. Mie scattering has an effect at wavelengths up to, or even greater than 10 μm [9, p. 116]. Natural and artificial aerosols are the primary causes of Mie scattering, which leads to Mie scattering being significant in overcast conditions [1, p. 99], [3, p. 10], [9, p. 116]. Mie scattering is the cause of a red sky following a forest fire [4, p. 2.16].

Non-selective scattering occurs when the wavelength of the propagating radiance is smaller than the physical size of the atmospheric molecule with which it is interacting [1, p. 99], [3, p. 10], [4, p. 2.16]. Non-selective scattering has very little dependency on wavelength, which results in radiance in the visible, near infrared (NIR) and midwave infrared (MWIR) regions all being equally affected [1, p. 102], [3, p. 10], [4, p. 2.16]. Non-selective scattering causes clouds and fog to appear white [3, p. 10], [4, p. 2.16].

Atmospheric turbulence: Atmospheric turbulence is a result of random variations in the atmospheric index of refraction, in both space and time [11, p. 444]. The change in the refractive index is due to changes in the atmospheric temperature, humidity and pressure [9, p. 97], [11, p. 444]. The varying refractive index causes deviations in the propagating wave front, which result in effects such as scintillation, beam steering, image dancing and blurring [1, p. 566], [9, p. 97], [11, p. 444].

Standard atmospheric models are used to define the atmospheric gas composition, temperature, pressure and relative humidity, from the earth to space. The 1976 United States (U.S.) Standard Atmosphere is an example of a standard atmospheric model. Standard atmospheric models are used in atmospheric radiative transfer codes (RTCs), such as Moderate Resolution Atmospheric Transmission (MODTRAN) [1, p. 293].

The vertical structure of the atmosphere is modelled as a set of horizontal layers, divided according to a property of the atmosphere, such as the change in temperature with altitude. The atmospheric layers defined by temperature are known as the troposphere, stratosphere, mesosphere, thermosphere and exosphere. These layers are separated by the tropopause, stratopause, mesopause and thermopause boundary layers [1, p. 292], [10], [28], [29]. The atmospheric layers do not have distinct boundaries, since the size of the layers fluctuates due to factors such as the time of day and the season of the year [29].

Figure 2.3 plots the 1976 U.S. Standard Atmosphere temperature, pressure and density vertical structures for each atmospheric layer, defined by the atmospheric temperature.

The atmospheric layers below 100 km contain more than 99% of the total mass of the earth's atmosphere, where approximately half of the atmosphere is contained within the first 5.6 km of the troposphere [9, p. 109], [10], [28], [29]. The troposphere contains

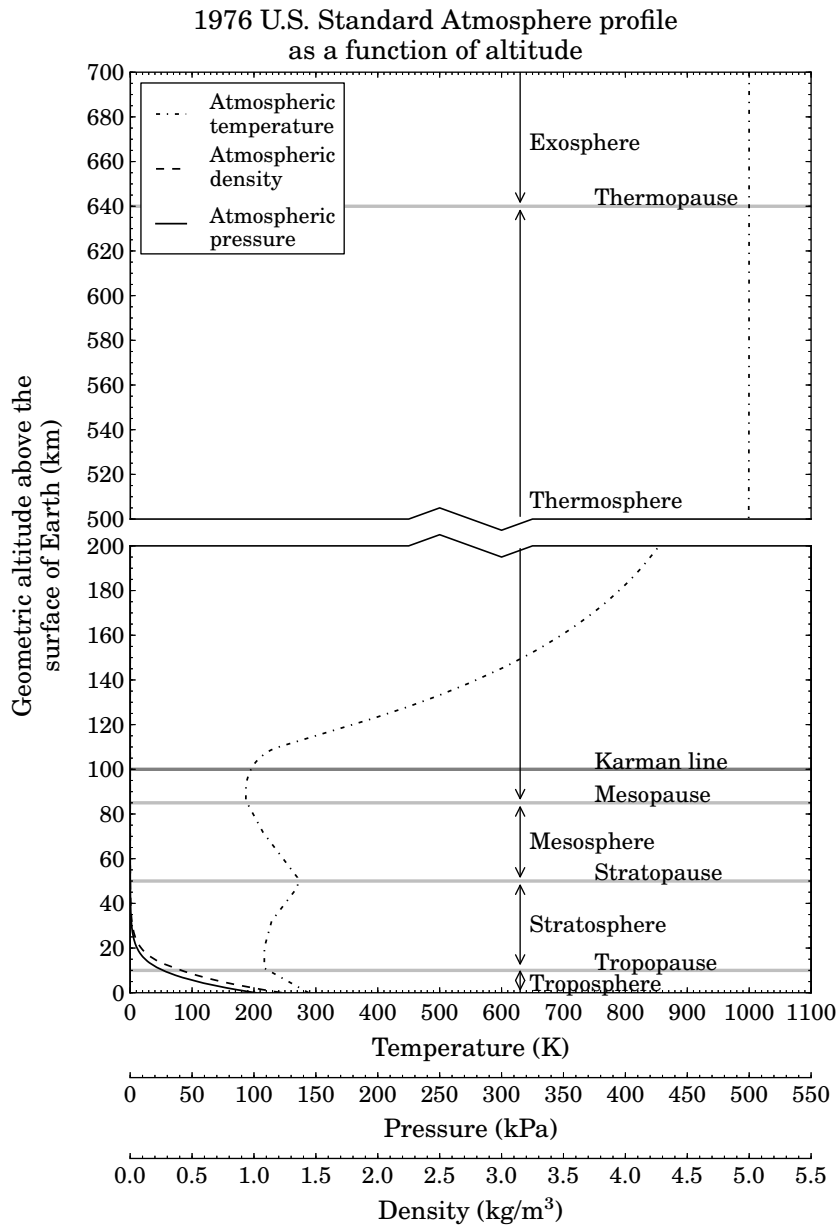


Figure 2.3: The vertical structure of the atmosphere, as defined by temperature, alongside the 1976 U.S. Standard Atmosphere temperature, pressure and density vertical structures (data sourced from [30]).

all of the weather activities of the earth [10], [28], [29]. Many remote sensing satellites that orbit the earth are found in the exosphere [10], [28]. The exosphere extends up to 10 000 km, where the atmosphere merges into space [28], [29].

There is no absolute dividing line between space and the earth's atmosphere [31]. The Aeronautic and Astronautic community, however, has set the top of the atmosphere to be at 100 km. This boundary is known as the Karman line and is used by scientists to study the balance of incoming solar radiance and outgoing reflected and thermal radiance, on the earth [31], [32].

2.2.1.3 The target

A target is any man-made or natural object imaged by a sensor. Propagating flux incident on the surface of a target will be absorbed, transmitted and reflected [3, p. 12], [4, p. 2.20], [9, p. 27], [14]. The absorbed flux is converted into another form of energy, such as thermal energy, which is emitted back into the atmosphere [1, p. 74], [4, p. 2.21]. Figure 2.4 illustrates the interaction between a target and incident optical flux.

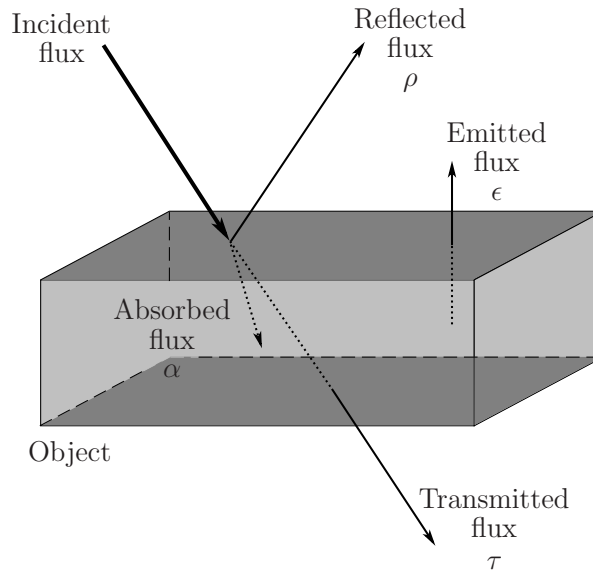


Figure 2.4: Interaction between incident optical flux and the surface of an object (adapted from [4, p. 2.21]).

The proportion of incident flux that is absorbed, transmitted and reflected by an object, depends on the physical characteristics of the flux, as well as the material type and condition of the object [3, p. 13], [4, p. 2.20]. For example, opaque objects have a zero transmittance since the majority of the incident flux is reflected and absorbed [1, p. 74], [3, p. 362], [9, p. 70], [14]. The absorption and emissivity of an object in thermal equilibrium is set to be numerically equal, in accordance with Kirchhoff's law [1, p. 74], [3, p. 362], [9, p. 69]. According to the principle of conservation of energy, the relationship between absorbed ($\Phi_{\text{absorbed}}(\lambda)$), transmitted ($\Phi_{\text{transmitted}}(\lambda)$) and reflected ($\Phi_{\text{reflected}}(\lambda)$) flux, given in watt, is expressed as [1, p. 74], [3, p. 362], [9, p. 27], [14],

$$\Phi_{\text{incident}}(\lambda) = \Phi_{\text{absorbed}}(\lambda) + \Phi_{\text{transmitted}}(\lambda) + \Phi_{\text{reflected}}(\lambda), \quad (2.2.1a)$$

which can be rewritten as a fraction of the total incident flux ($\Phi_{\text{incident}}(\lambda)$),

$$\alpha(\lambda) + \tau(\lambda) + \rho(\lambda) = 1, \quad (2.2.1b)$$

where $\alpha(\lambda)$ is the fraction of absorbed flux, $\tau(\lambda)$ is the fraction of transmitted flux and $\rho(\lambda)$ is the fraction of reflected flux. All components in (2.2.1a) and (2.2.1b) are a

function of wavelength. The measured reflectance is also dependent on the illumination and viewing angles [1, p. 121], [14].

The reflectance properties of objects are an important factor when calculating the at-sensor radiance in the visible region [2, p. 13], [3, p. 13]. When comparing objects in an image, both the spectral reflectance and the geometric manner in which the objects reflect radiance must be considered [3, pp. 13 and 24].

Spectral characteristics: Insight into the spectral characteristics of an object is gained by studying the spectral reflectance curve of that object [3, p. 13]. A spectral reflectance curve is defined as the percentage of incident radiance reflected by an object at each wavelength across the EM spectrum [4, p. 2.24].

Spectral reflectance curves aid in distinguishing different types of materials from one another, since the curves differ between materials, as well as within a material type [2, p. 13], [3, p. 13]. The spectral reflectance curves of objects of the same material type will exhibit similar general characteristics [2, p. 15]. Spectral reflectance curves also influence the choice of wavelength range to use when acquiring data regarding an object [3, p. 13], [4, p. 2.27]. For instance, the NIR region can be used to distinguish easily between vegetation, soil and water, as shown in Figure 2.5 [4, p. 2.27].

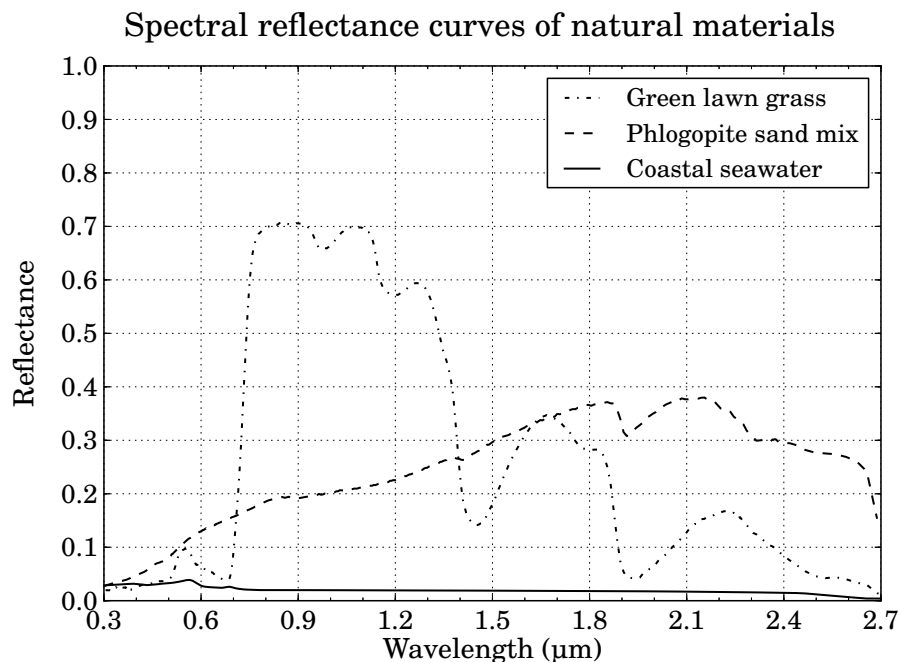


Figure 2.5: Comparison of spectral reflectance curves of grass, sand and seawater (adapted from [4, p. 2.27] and using data from [33]).

Directional characteristics: The directional characteristics describe the geometric manner in which an object reflects radiance. The directional distribution of the reflected radiance is defined as being specular or diffuse or a combination of both. The type of

reflectance that will take place is determined by the surface roughness of an object and the wavelength range of the incident radiance [1, p. 122], [3, pp. 24-25]. Figure 2.6 illustrates the diffuse, specular and mixed reflectance distributions.

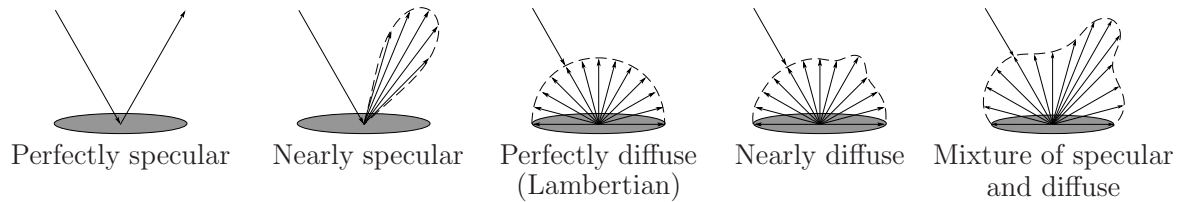


Figure 2.6: Diffuse, specular and mixed reflectance distribution of objects (adapted from [1, p. 122] and [3, p. 25]).

Perfectly specular reflectors are objects with flat surfaces that behave like mirrors, where the angle of reflection is equal to the angle of incidence [1, p. 122], [3, p. 25], [4, p. 2.22]. Nearly specular reflectors will reflect radiance in a cone shape around the ideal specular reflection ray [1, p. 123], [9, p. 76]. Perfect diffuse, or Lambertian reflectors, are objects with rough surfaces that reflect radiance uniformly in all directions, regardless of the source or direction of the incident radiance [1, p. 123], [3, p. 25], [4, p. 2.22], [9, p. 76]. Nearly diffuse objects will appear to reflect more radiance in the ideal specular direction [1, p. 123]. Most real world objects have mixed reflectance characteristics which result in the object appearing brighter in the direction of ideal specular reflection and backscatter and darker when viewed at the grazing angles [1, p. 123], [3, p. 25], [9, p. 82].

The object's geometric reflectance properties are represented by a bidirectional reflectance distribution function (BRDF) [3, p. 27]. BRDF models are used to express deviations in the reflectance geometry of real materials from the Lambertian model [2, p. 53]. BRDF models for the visible and NIR regions include the Phong, Ward, Lafortune, Cook-Torrance and Torrance-Sparrow models [1, p. 124], [9, p. 82].

2.2.1.4 The sensor and satellite

The sensor is one of the most complex links in the image chain [1, p. 153]. Sensors used in remote sensing systems are classified as being either passive or active sensors [15, p. 9]. Passive sensors measure flux generated by an external source, which is reflected or emitted towards the sensor from the target. Active sensor systems generate radiation that is beamed down to the target. The sensor then measures the portion of flux reflected towards the sensor from the target [1, p. 7], [2, p. 10], [15, p. 9]. Both active and passive sensors are classified as being either imaging or non-imaging sensors, according to whether or not an image is produced. Imaging sensors are further classified as being optical or non-optical, according to whether or not an optical system is present in the sensor [15, p. 9]. The

class of optical sensors is divided into film-based and digital cameras. Digital cameras are further described by the image acquisition method employed, namely, scanning over the scene or taking a snap-shot frame of the scene [15, p. 9]. This study focused on passive imaging sensors that use optics and a digital camera.

The sensor converts the collected at-sensor radiance into an image of the observed scene [2, p. 75]. The sensor components affect the radiometric quality of the image [2, p. 110]. The properties of the sensor platform's orbit and attitude, the sensor scanner and the rotation and shape of the earth affect the geometric characteristics of the image [2, p. 110]. The sensor system is described by the sensor components and scanning configuration and the satellite platform and orbit.

Sensor components: The primary components of an electro-optical sensor system include a scanner, imaging optics, filters or dispersion elements, detectors, electronics and a means to record data. The sensor electronics include the preamplifier and analogue-to-digital converter (ADC) [1, pp. 167-168], [2, p. 76].

The electro-optical system makes use of a scanning operation to convert the spacial at-sensor radiance into a continuous, time-varying optical signal [2, p. 76]. The imaging optics are then used to transfer the optical signal to the detector focal plane [2, p. 104], [9, p. 221]. The imaging optics consist of mirrors, also referred to as reflective optics, which reflect all wavelengths in the EM spectrum equally. There are several long focal length reflective optic designs, such as the Cassegrainian type, commonly used in remote sensing systems [1, pp. 165-166]. If a wide FOV is required, then a three-mirror anastigmatic (TMA) design can be used [1, p. 166].

Before the detectors collect the optical flux, the flux is separated into different spectral components. Multispectral filters or dispersion elements, such as prisms, are used to split non-thermal flux into the different spectral components [2, p. 104], [3, p. 326]. A dichromatic grating is used if thermal and non-thermal flux must be separated from one another [3, p. 326].

The detectors convert the signal into a continuous time-varying electronic signal [1, p. 167], [2, p. 76], [9, pp. 221-222]. The electronic signal is amplified and processed by the sensor electronics [2, p. 76]. An ADC is used to sample the processed signal in time and to quantise the signal into discrete DN values [2, p. 76]. The digital data are then recorded and transmitted [1, pp. 167-168].

The image acquisition processes for an across-track and an along-track scanning electro-optical sensor system are illustrated by the flow diagrams in Figures 2.7 and 2.8 respectively. Both scanning systems follow the same image acquisition process, with the exception of the scanner component. An across-track scanner makes use of a mechanical scanning mirror, such as a rotating or oscillating mirror, to image the scene. An

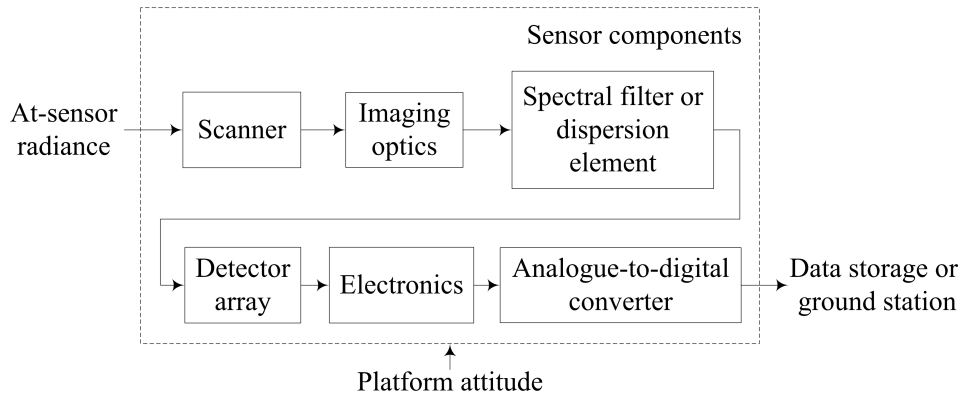


Figure 2.7: The image acquisition process of an across-track scanning electro-optical sensor (adapted from [2, p. 76]).

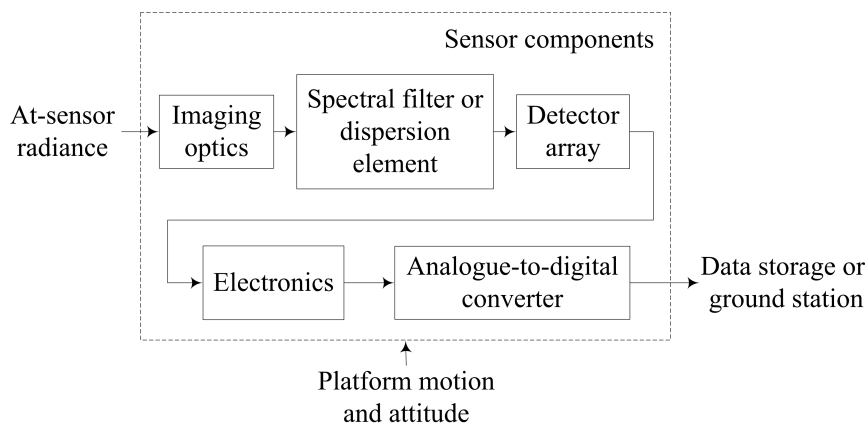


Figure 2.8: The image acquisition process of an along-track scanning electro-optical sensor (adapted from [2, p. 76]).

along-track scanner makes use of the platform's motion to provide the scanning action [3, pp. 326 and 332].

Sensor scanning configurations: The sensor scanner is set to be either an along-track or across-track type scanner. Sensor scanning configurations include the line, whiskbroom, paddlebroom and pushbroom scanner configurations and the framing two-dimensional (2D) array configuration [1, pp. 194-212], [2, pp. 19-20], [15, pp. 18 and 20]. Figure 2.9 depicts the different sensor scanning configurations.

Whiskbroom scanners, also known as across-track scanners, image a scene by using oscillating scan mirrors to scan over the area in the across-track direction [2, p. 19], [3, p. 326], [15, p. 20]. An oscillating scan mirror scans across the track as the sensor platform advances slightly in the along-track direction; in so doing, consecutive image lines are formed along the track. The oscillating scan mirror causes the across-track scan direction to reverse for each scan cycle [1, p. 194], [2, p. 19], [15, p. 20]. Several detector elements are aligned along the track so that parallel across-track lines are scanned for each scan mirror

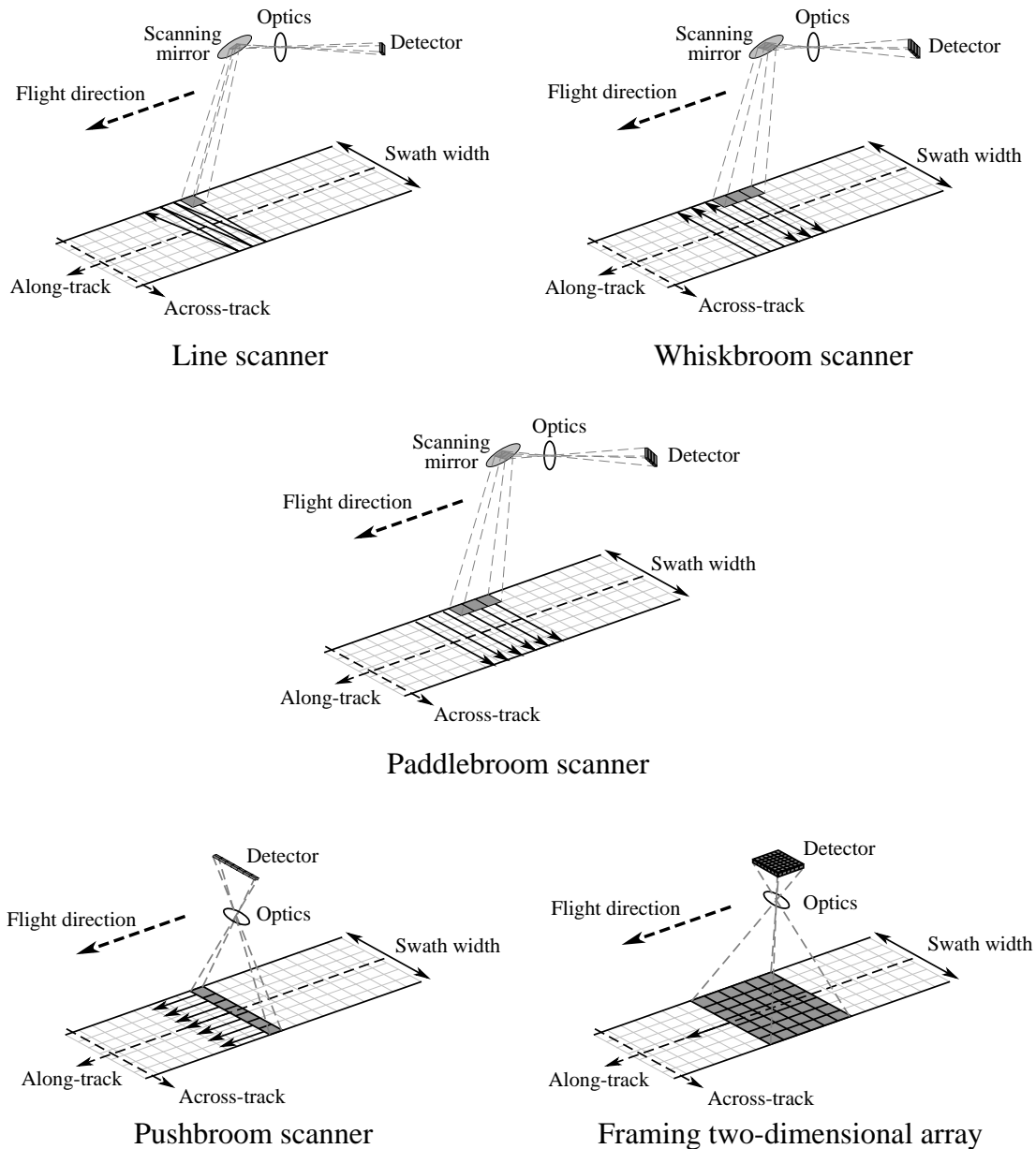


Figure 2.9: Different sensor scanning configurations used to image a scene (adapted from [2, pp. 19-20] and [15, pp. 18, and 20]).

cycle [1, p. 200], [2, p. 19]. Line scanners operate in the same manner as whiskbroom scanners, except that the line scanner uses only a single detector element to scan the scene [2, p. 19].

Paddlebroom scanners are similar to whiskbroom scanners, except that the paddlebroom scanner uses a two-sided scanning mirror, which rotates 360°, to scan across the track. The rotating mirror allows for each scan cycle to be in the same direction across the track [2, p. 19].

Pushbroom scanners, also known as along-track scanners, use a linear array of detectors to scan over the scene in the along-track direction as the satellite platform moves along the track [2, p. 19], [3, p. 326]. The linear array of detector elements is aligned perpendicular

to the direction of the platform motion [2, p. 19]. In this manner an entire line of data that covers the full ground swath width is collected with each scan [1, p. 207], [2, p. 19]. Pushbroom sensors have no moving parts and have long dwell times, which make them suitable for implementation in satellite remote sensing systems [1, p. 208].

Framing 2D array imagers consist of a 2D array of detector elements and a mechanical or electronic shutter to capture images of a scene in the same manner as a conventional film camera [1, p. 211], [15, p. 18]. As the platform on which the sensor is mounted moves along the track the sensor takes snapshots of the scene.

The detectors used in the whiskbroom, paddlebroom and pushbroom configurations form a linear array of charge coupled devices (CCDs) in the sensor focal plane [3, p. 333], [15, p. 13]. Each detector in the array is dedicated to imaging a single row or column of data [3, p. 333]. Figure 2.10 illustrates different CCD line configurations.

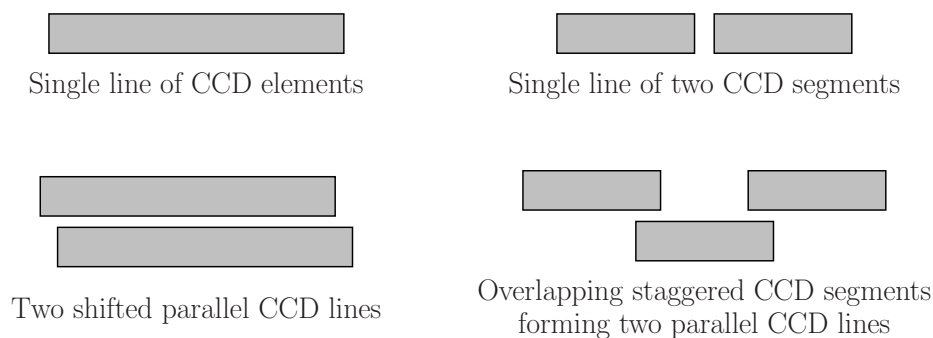


Figure 2.10: Possible configurations of the charge coupled device lines, in the sensor focal plane (adapted from [15, p. 13]).

The CCD array of detectors forms a continuous line that can be segmented, shifted or staggered or a combination of these [1, p. 208], [15, p. 13]. Each CCD segment is referred to as a detector sub-array (DSA). The detectors are shifted and staggered by a fraction of the pixel size and detectors in adjacent staggered DSAs may overlap [15, p. 13].

A short vector of detector elements can be aligned parallel to each other along the image scan direction. This results in the signal from successive detectors along the same detector scan line being delayed, which allows the same point in a scene to be imaged multiple times. This detector array configuration is used for time delay integration (TDI). The image is sampled by the detectors in the first line and then moments later by the next line of detectors, as the sensor sweeps over the image. The sampled signals from the group of detectors in the same scan line are summed together and averaged to increase the signal levels and to reduce the noise for the image pixel at that point in the scene. This allows even faint targets to be detected [1, p. 597], [34]. The charge is then read out and transferred to a shift register at a rate that matches the platform velocity [15, p. 16], [34]. Figure 2.11 illustrates TDI using a three stage TDI detector.

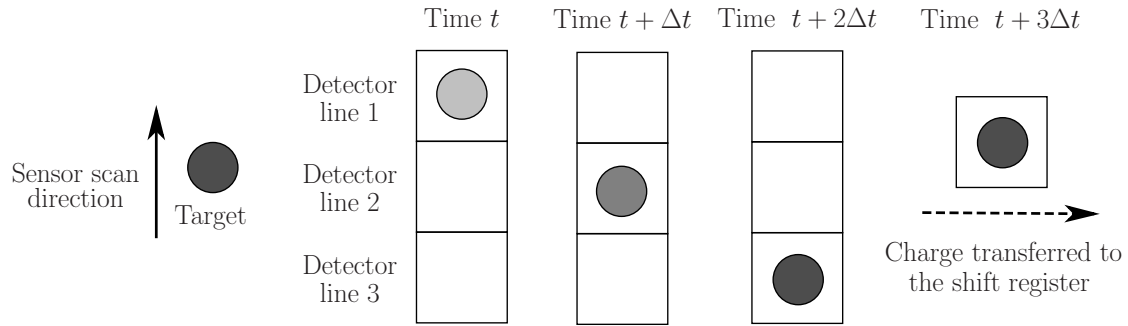


Figure 2.11: Illustration of time delay integration (TDI) process using a three stage TDI detector (adapted from [15, p. 16] and [34]).

Satellite platform: In order for an earth observation sensor to collect the reflected or emitted radiance from a target, the sensor must be mounted on a stable platform removed from the target [8, p. 34]. The general architecture of an earth observation satellite platform is divided into two main components, namely, the payload and the bus [35]. The payload consists of observation instruments used for remote sensing purposes. The satellite bus supports the payload mass and consists of instruments to store and transmit data, provide electrical power to the payload and control the payload temperature, attitude and orbit [35], [36].

Satellite orbit: A satellite orbit must match the capability and objective of the satellite sensor [8, p. 36]. Two common types of satellite orbit are the low earth orbit (LEO) and the geo-synchronous earth orbit (GEO), illustrated in Figure 2.12 [1, p. 49].

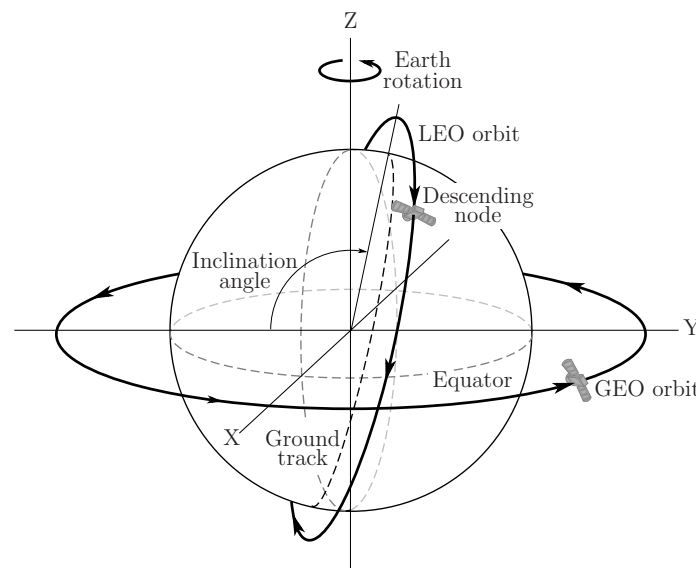


Figure 2.12: Two common orbits for earth observation remote sensing satellites (adapted from [1, pp. 50 and 52] and [2, p. 116]).

GEO orbits are commonly used by communication satellites [16, p. 163]. A satellite in a GEO orbit revolves around the earth at the same rate as the earth rotates, this allows a satellite to remain roughly over a fixed longitude position [1, p. 49], [10], [16, p. 163].

LEO orbits are commonly used by remote sensing satellites and are often sun-synchronous, near-polar, descending node orbits [1, p. 49], [2, p. 115], [10], [16, p. 163]. Sun-synchronous orbits allow a satellite to maintain a constant sun-earth-sensor angle, since the satellite will always pass over a location on the earth at the same local solar time [1, p. 49], [8, p. 36], [10]. Satellite orbits are defined as being equatorial, polar, prograde or retrograde [16, p. 157]. A satellite in an equatorial orbit will remain over the equator, whereas, a satellite in a polar orbit will travel over the poles. Prograde and retrograde orbits travel in an easterly or westerly direction around the earth, respectively. Prograde and retrograde orbits are near-polar orbits if the orbit inclination angle is close to 90° [10], [16, p. 156]. The direction in which the satellite travels along an orbit is described as being either ascending or descending. Satellites in a descending node orbit cross the equator from North to South, on the sunlit side of the earth. Satellites in an ascending node orbit cross the equator from South to North, on the sunlit side of the earth [2, p. 115].

2.2.2 Transmission and reception of the recorded data

A satellite stores the collected data on-board until it comes within range of a receiving station, at which point the data are transmitted to the station [4, p. 2.31]. Data are converted to DNs, using an ADC, and then attached to a signal before being transmitted. This is done in order to maintain the quantitative integrity of the data during transmission [1, p. 53], [4, p. 2.31].

2.2.3 Processing, interpreting and analysis

At the receiving station, the carrier signal is filtered from the transmitted data. The data are then decoded, recorded and made available for further processing [4, p. 2.31]. Geometric registration and calibration is performed on the data so that it can be compared to other sets of data. Feature extraction is also performed on the data [26]. The processed image is then interpreted visually or digitally in order to extract information about the imaged target [3, p. 2], [8, p. 6], [26].

2.2.4 Application of the data

The processed data are used in applications in order to better understand a specific target or to reveal new information about a target [8, p. 6]. The data can also be used in a decision-making process when solving a specific problem [3, p. 2], [8, p. 6].

2.3 Existing remote sensing simulation models

There are several existing models in use to simulate remote sensing systems. The Optronics System Simulator (OSSIM) simulation environment is one such example. The OSSIM environment is primarily used to simulate missile seeker sensors and thermal imaging systems [6]. Examples of simulation models used to simulate satellite remote sensing systems include Digital Imaging and Remote Sensing Image Generation (DIRSIG), Parameterized Image Chain Analysis and Simulation Software (PICASSO) and simulation model Software Environment for the Simulation of Optical Remote Sensing Systems (SENSOR) [1, p. 626], [17], [37].

2.4 Conclusion

A satellite remote sensing system describes the remote sensing process followed by an earth observation satellite. The remote sensing process follows the process of data acquisition, transmission, reception, analysis and implementation. The main focus of this study was on the data acquisition processes. The data acquisition process consists of four main elements, namely, EM radiation, the atmosphere, the target and the sensor. Each of these elements affects the quality of the sensor output and these are discussed in this thesis.

Remote sensing of EM radiation is described in terms of the spectral range used when collecting data. The different forms of EM radiation provide discrete information about the target from which it is reflected or emitted. EM radiation is transmitted through the atmosphere only in atmospheric transmission windows. Atmospheric turbulence, scatter and absorption affect the intensity and spectral composition of the radiance reaching the sensor by attenuating, amplifying or increasing the propagating radiance. The target is any man-made or natural object of interest. The target reflectance properties, both spectral and geometric, are an important factor when collecting data in the visible region.

The sensor is one of the most complex links in the image chain. This study focused on passive imaging sensors that use optics and a digital camera. The primary components of such a sensor include a scanner, imaging optics, filters or dispersion elements, detectors, electronics and a means to record data. The sensor scanner is either an along-track or across-track type scanner. The sensor makes use of a linear array of detectors that form a continuous line that can be shifted or staggered, or a combination of these. Furthermore, the satellite orbit used must match the capability and objective of the satellite sensor. LEO orbits are commonly used by remote sensing satellites and are often sun-synchronous, near-polar, descending node orbits.

Chapter 3

Design of a Satellite Remote Sensing Software Model

3.1 Introduction

As discussed in Chapter 2, there are a number of factors that influence the radiometric and geometric quality of the output of a satellite remote sensing system. This study focused on the image capturing process, specifically the target-atmosphere-sensor relationship, in a passive satellite remote sensing system. A satellite remote sensing system software model was designed to model this relationship. To model the target-atmosphere-sensor relationship accurately, the sensor architecture, satellite motion, atmospheric effects and the target properties are considered.

The target properties in the model were calculated using a set of user-created test targets and a BRDF model implemented in a radiometric signature rendering equation. The signature rendering equation was used to model the effect that each of the factors in the target-atmosphere-sensor relationship has on the at-sensor radiance and, therefore, the output image data. The atmospheric effects are included in the model by using the MODTRAN4 RTC and an adjacency effect atmospheric aerosol point spread function (PSF). The sensor architecture is accounted for by means of a generic passive optical satellite sensor model. This generic satellite sensor model provides a framework that can be used to represent a real world satellite sensor, by selecting appropriate values. The CCD array, optics and image sampling method form the core of the sensor model. In order to simulate the satellite motion, the capability to specify and calculate a representative satellite orbit trajectory is included in the model.

Satellite sensor detector elements are normally placed in a staggered configuration to achieve a 100% fill factor without touching each other. The spatial stagger, combined with temporal signal sampling, results in geometric discontinuities in the sampled image which makes visual analysis of the image difficult. For this purpose the sensor model

includes an image correction capability, to correct the geometric discontinuities in the sensor output image.

The OSSIM simulation environment was made available to the author to implement the satellite remote sensing system model. OSSIM was originally designed for short-range, low altitude electro-optical system simulation. OSSIM provided the infrastructure and key capabilities upon which this specialist satellite remote sensing system work built. For example, the OSSIM simulation environment includes a staring array sensor model which was adapted, and to which additional functionality was added, in order to function as a generic satellite sensor model. In this document, the software implementation of the satellite remote sensing system model is referred to as OSSIM for space applications (OSPA). The OSPA model is implemented as an OSSIM application model and new functionality was implemented in the OSSIM core library. A description of the OSSIM simulation environment and code structure is provided in Appendix A.

All elements of the generic satellite model are implemented in C++ in the OSPA environment. The model's input parameters are encoded in a tree of Extensible Mark-up Language (XML) files. Each file contains the parameters for a specific component. All the XML files that are used to describe a single simulation scenario are grouped together and are referred to as a 'test point'. Python code is used to plot and evaluate the outputs.

Each object in the OSPA environment must include a three-dimensional (3D) mesh model of the object body. The 3D mesh model is necessary in order for sensors to view the object. Appendix B provides a description of the satellite body 3D mesh model. Appendix C provides a description of the 3D mesh models of the different test targets.

To render an image in the OSPA environment, the OSSIM core library is used to map the 3D objects into a 2D image plane [7, p. 49]. Jagged edge anti-aliasing and radiometric accuracy is achieved by rendering the image plane in a high spatial resolution. This high resolution image includes the scene content and atmospheric effects. The OSSIM application module is used to apply sensor artifacts and degradation effects to the image, which is then resampled to produce a lower resolution sensor image [7, pp. 54 and 57]. A description of the high resolution and sensor images is given in Appendix A.4.

3.2 Functional diagram for the sensor image capturing process

The approach to implementing the model is driven by a functional analysis. The functional block diagram identifies key elements and defines how the main steps in the image capturing process are connected and implemented in OSPA. Each step is represented by a functional unit (FU) block as shown in Figure 3.1.

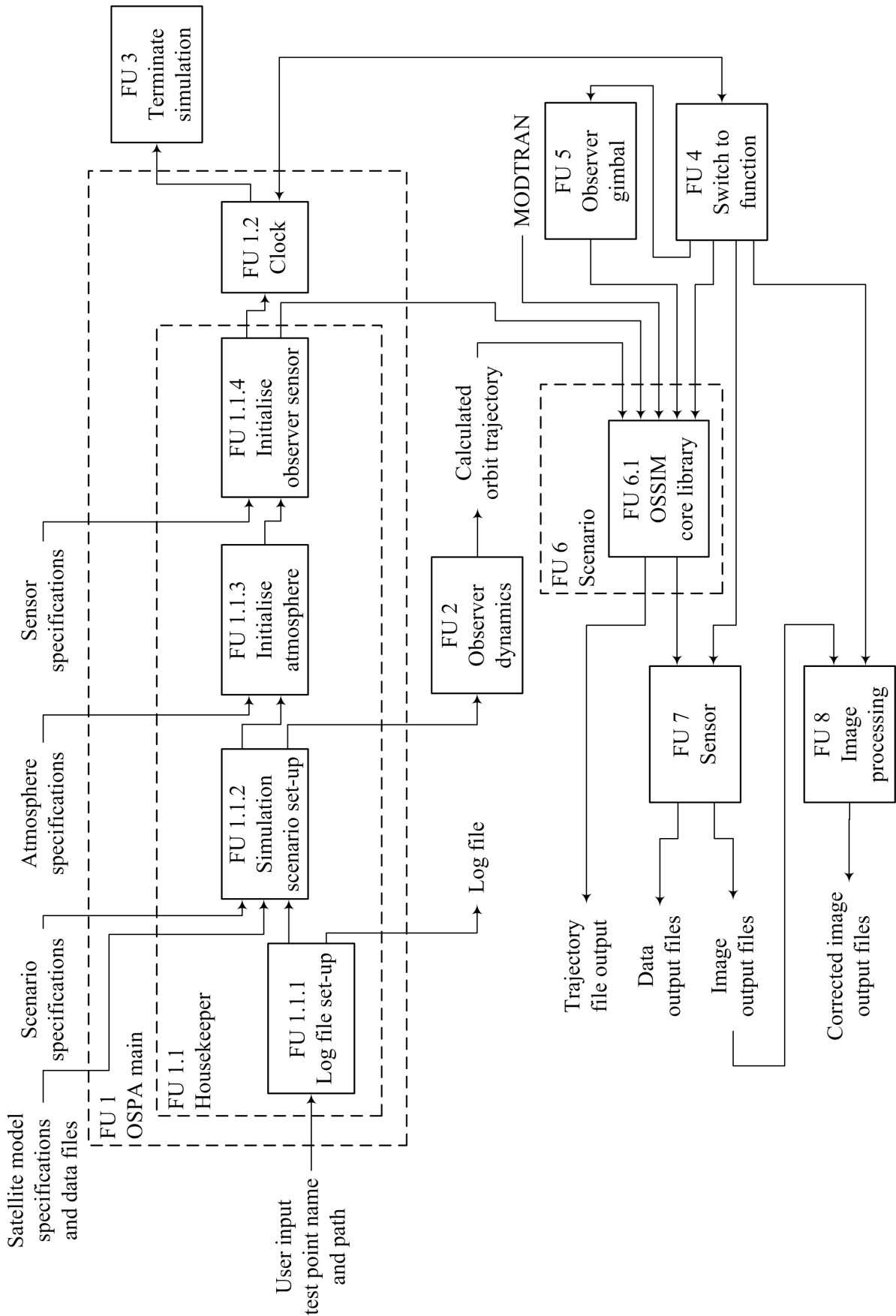


Figure 3.1: Functional diagram for OSPA remote sensing satellite image capture.

As seen in Figure 3.1, OSPA can be divided into eight main FUs. The first FU is the OSPA main file which is used to control the simulation run. FU 1 is divided into the housekeeper FU (FU 1.1) and the clock FU (FU 1.2). The housekeeper manages the set-up of the simulation and the clock manages the simulation run-time. The first input to FU 1 is the test point file path.

FU 1.1 manages a number of the simulation set up functions. The main set-up functions include setting up a log file (FU 1.1.1), the simulation scenario (FU 1.1.2), the atmosphere (FU 1.1.3) and the observer sensor (FU 1.1.4). Included in the simulation scenario set-up is the calculation of the observer orbit trajectory which is calculated using the OSPA observer dynamics FU (FU 2). To set up the simulation, FU 1.1 requires the test point XML and data files. The test point XML files describe the satellite sensor configuration, the scanning method, image correction settings, the satellite orbit specifications, the atmosphere and the targets being imaged. The test point data files include the polygon model of the object body, the material database file, spectral files and the atmospheric aerosol scattering PSF. FU 1.1 outputs a log file, describing the simulation status and warning and error messages. The output of FU 1.1 is used to add the targets and sensor objects to the simulation for rendering. The objects are added to the simulation using the OSSIM core library functionality in FU 5.1 [38, p. 3-1-1].

With each simulation time increment FU 1.2 checks the status of the termination flag. If the simulation end time has been reached, then the simulation is terminated by FU 3. FU 3 sets the simulation termination flag to be true and clears the memory. If the simulation end time has not been reached then FU 4 is executed. The simulation execution moves back and forth between FU 1.2 and FU 4 until the simulation is terminated.

FU 4 is used to determine which function to execute based on the current simulation time. These functions include the OSPA specific functionality (FUs 5, 7 and 8) and the OSSIM core library functions (FU 6.1), accessed via the scenario class (FU 6).

FU 5 is used to calculate the attitude angles of the satellite gimbal, which simulates the scan mirror attitude. FU 5 is only executed if the satellite gimbal is activated. The attitude angles of the satellite gimbal are recorded for the observer object in FU 6.

FU 6 acts as a global access point to the OSSIM core library (FU 6.1) [38, p. 3-8-3]. FU 6.1 handles OSSIM functionality such as the scene rendering, atmospheric and radiometric calculations and the integration of all modules required to work together [7, p. 26]. FU 6.1 uses MODTRAN to perform the atmospheric calculations.

The satellite observer's trajectory information and scan mirror attitude for the current simulation time is accessed via FU 6. This information is used to set the line-of-sight of the sensor and is written to a trajectory file.

FU 7 is used to image the targets, using the OSSIM and OSPA functionalities. The image data are written to an image file and to radiometric and statistic data files.

FU 8 is used to perform basic image processing on the output images produced by

FU 7. The image processing corrects for geometric discontinuities. FU 8 is only executed if the image processing functionality is activated.

FU 2, 5, 7, 8 and the radiometric signature rendering equation in FU 6.1 are discussed in further detail in this study.

To test the satellite remote sensing model, a real world satellite (WorldView-2) was modelled using the OSPA environment.

3.3 Motivation for simulating the satellite image capturing process

Simulation models that are able to accurately simulate the output of a satellite sensor, are valuable remote sensing application tools. A satellite remote sensing simulation tool would be useful to a satellite sensor design team, as well as to the image scientists who process and analyse satellite data.

Some applications for a satellite remote sensing simulation tool are listed.

1. Simulation models aid in determining system requirements, solving design problems, reducing risks, costs and development time when designing a real world satellite sensor [5], [6].
2. The simulation model aids in the optimising and in predicting the performance of a designed satellite sensor [7, p. 25].
3. The simulation model allows a developer to see how the satellite sensor, environment and target will interact with one another. This interaction can be viewed for a number of different scenarios and environmental conditions [6], [7, pp. 25 and 92].
4. The impact that the sensor design has on an image can be studied, using the simulation environment [1, p. 641].
5. Simulations are used to reduce the dependency on extensive field tests when developing, testing and calibrating sensors [7, p. 25].
6. The simulation model can be used to help develop and evaluate image analysis and processing algorithms for different sensors [1, p. 641], [5].
7. The simulation model can be used to generate a large set of training material that covers a wide range of user defined conditions and scenes [7, p. 39]. This saves on the cost and difficulty of acquiring real world satellite data.
8. The simulation model can be used to design test targets to be viewed by a satellite, for a specific application.

3.4 Conclusion

A model of the image acquisition process in a satellite remote sensing system was designed and implemented in C++ code. The software implementation of this model is referred to as OSPA. OSPA focuses on the target-atmosphere-sensor relationship. The model includes a set of test targets, the atmosphere, and a generic satellite sensor model that serves as a framework to represent a real world satellite sensor. A signature rendering equation is used to model the effect that each of the components in the model have on the at-sensor radiance. The model is built around and expands on the OSSIM capabilities.

The OSPA simulation tool can be used when designing, optimising and testing real world satellite sensors. The simulation tool can also be used to generate training data for a number of different conditions, scenes and sensor types. Such data are useful when evaluating image analysis and processing algorithms.

The WorldView-2 satellite was chosen to be modelled in the OSPA environment in order to test the satellite remote sensing model.

Chapter 4

The Signature Rendering Equation

4.1 Introduction

A signature rendering equation models the total at-sensor radiance and the effect of the sensor response on the collected radiance. Each term in the signature rendering equation describes a different contributor to the signature. The contribution of each term is affected by the path that the radiance travels through the atmosphere. The propagation paths found in passive remote sensing describe the paths followed by radiance originally emitted by the sun and by thermal radiation emitted from objects in the scene [1, p. 57]. The signature rendering equation typically includes terms to describe the paths followed by the self-emitted thermal radiance of the target, reflected sunlight, reflected skylight and atmospheric path radiance [2, pp. 47-49 and 63], [3, pp. 23-24], [6]. Additional terms can be added to the equation to meet the user's requirements for a specific application.

The signature rendering equation terms account for various factors that contribute to the at-sensor radiance [9, p. 281]. Contributing factors include the atmosphere, the source-target-sensor geometry, the target orientation, the surrounding objects and terrain and the optical properties of the target [1, pp. 112 and 117], [3, p. 24], [6], [17]. Almost all of the factors are wavelength dependent [3, p. 24], [9, p. 281]. Each term in the equation is weighted by the sensor spectral response [6], [9, p. 281].

The signature rendering equation can be simplified, based on the spectral range over which the at-sensor radiance is modelled. For modelling at-sensor radiance in the visible and NIR regions, the equation is simplified to include only the terms that describe the solar radiance contributions. In the visible and NIR regions the solar at-sensor radiance is orders of magnitude higher than the thermal radiance from terrain objects with temperatures around 300 K. The contribution to the at-sensor radiance of the thermal radiance can therefore be considered to be negligible in the shorter wavelength bands [1, p. 145]. Similarly, for modelling at-sensor radiance in the longwave infrared (LWIR) regions, the equation can be simplified to include only the terms that describe the thermal radiance

contributions [1, p. 147]. If the at-sensor radiance is modelled for the MWIR region, then all terms describing the solar and thermal radiance contributions must be included in the equation [1, p. 147], [9, p. 299].

The OSPA signature rendering equation was designed by expanding on the OSSIM signature rendering equation to include an adjacency effect term required when modelling the at-sensor radiance for a satellite sensor.

4.2 The satellite signature rendering equation

A comprehensive satellite signature rendering equation must include radiance terms to describe the self-emitted thermal radiance of the target, the sunlight reflected off the target, reflected skylight, atmospheric path radiance and the ambient environment contributions. The trapping and adjacency effect should also be included in the satellite signature rendering equation. The signature rendering equation should take into account that both opaque and semi-transparent objects can be present in a scene. Semi-transparent objects that may be viewed by a satellite include smoke and thin cirrus clouds. The sensor response should be included in the signature rendering equation, in order to simulate an accurate representation of the recorded at-sensor radiance.

The OSSIM signature rendering equation includes most, but not all, of the terms required to model the at-sensor radiance for a satellite. An adjacency effect term, as discussed in section 4.3, must be added to the OSSIM equation in order to make the equation applicable for use with a satellite model [6], [9, p. 281].

The OSSIM signature rendering equation is given in (4.2.1) [6], [9, p. 281]. This equation is not mathematically rigorous and serves only to define various contributors to the at-sensor radiance, $L_{\Delta\lambda}$ ($\text{W}/(\text{m}^2 \cdot \text{sr})$), in a concise manner.

$$\begin{aligned}
 L_{\Delta\lambda} = & \overbrace{\Delta\rho_o \psi_s \cos\theta_s \int_{\lambda_0}^{\lambda_1} \epsilon_{s\lambda} L_\lambda(T_s) \tau_{so\lambda} f_o(\theta_s, \theta_v, \varphi_s, \varphi_v) \tau_{a\lambda} S_\lambda \, d\lambda}^{\text{reflected sunlight}} \\
 & + \overbrace{\Delta\epsilon_o \int_{\lambda_0}^{\lambda_1} \epsilon_{o\lambda}(\theta_v, \varphi_v) L_\lambda(T_o) \tau_{a\lambda} S_\lambda \, d\lambda}^{\text{self-emitted thermal radiance}} + \overbrace{\int_{\lambda_0}^{\lambda_1} L_{\text{path}\lambda} S_\lambda \, d\lambda}^{\text{atmospheric path radiance}} \\
 & + \overbrace{\Delta\rho_o \cos\theta_a \int_{\lambda_0}^{\lambda_1} \int_{\text{sky}} L_{\text{sky}\lambda} \rho_{o\lambda} \tau_{a\lambda} S_\lambda \, d\Omega \, d\lambda}^{\text{diffuse reflected sky}} \\
 & + \overbrace{\Delta\tau_o \int_{\lambda_0}^{\lambda_1} \epsilon_{b\lambda} L_\lambda(T_b) \tau_{bo\lambda} \tau_{o\lambda} \tau_{a\lambda} S_\lambda \, d\lambda}^{\text{transmitted background}} \\
 & + \overbrace{\Delta\rho_o \int_{\lambda_0}^{\lambda_1} \int_{\text{amb}} \epsilon_{a\lambda} L_\lambda(T_a) \tau_{ao\lambda} \rho_{o\lambda} \tau_{a\lambda} S_\lambda \, d\Omega \, d\lambda}^{\text{diffuse reflected ambient environment}}.
 \end{aligned} \tag{4.2.1}$$

Table 4.1: Description of variables used in the OSSIM signature rendering equation [6], [9, p. 282].

Variable	Description	Variable	Description
$L_{\Delta\lambda}$	At-sensor spectral radiance in the wavelength band $\Delta\lambda$	$\Delta\epsilon_o$	Object spatial texture variation in emissivity
$L_{\lambda}(T_o)$	Spectral black body radiance at the object temperature T_o	$\tau_{o\lambda}$	Object spectral transmittance
$L_{\lambda}(T_b)$	Spectral black body radiance at the background temperature T_b	$\tau_{bo\lambda}$	Background-object path spectral atmospheric transmittance
$L_{\lambda}(T_a)$	Spectral black body radiance at the ambient environment temperature T_a	$\tau_{ao\lambda}$	Ambient environment-object path spectral atmospheric transmittance
$L_{\lambda}(T_s)$	Spectral black body radiance at the sun temperature T_s	$\tau_{so\lambda}$	Sun-object path spectral atmospheric transmittance
$L_{path\lambda}$	Spectral atmospheric path radiance (emitted and scattered radiance)	$\tau_{a\lambda}$	Object-sensor path spectral atmospheric transmittance
$L_{sky\lambda}$	Spectral sky radiance (emitted and scattered radiance)	$\Delta\tau_o$	Object spatial texture variation in transmittance
$f_o(\theta_s, \theta_v, \varphi_s, \varphi_v)$	Object surface bidirectional reflectance distribution function	θ_a	Angle between the surface normal and the vertical
$\rho_{o\lambda}$	Object surface diffuse spectral reflectance	θ_s	Zenith angle of the incident solar irradiance vector
$\Delta\rho_o$	Object spatial texture variation in reflectivity	θ_v	Zenith angle of the viewing direction vector
$\epsilon_{o\lambda}(\theta_v, \varphi_v)$	Object surface directional spectral emissivity	φ_s	Azimuth angle of the incident solar irradiance vector
$\epsilon_{s\lambda}$	Spectral emissivity of the surface of the sun	φ_v	Azimuth angle of the viewing direction vector
$\epsilon_{b\lambda}$	Background spectral emissivity	ψ_s	Scaling factor relating solar radiance to the object surface
$\epsilon_{a\lambda}$	Ambient environment spectral emissivity	S_{λ}	Sensor spectral response

Table 4.1 provides a description of each variable used in (4.2.1).

Figure 4.1 is a visual representation of the propagating radiance paths and the contributing factors described by the terms of the OSSIM signature rendering equation [9, p. 280]. Each term in (4.2.1) is represented by one or more of the propagating paths. The propagation paths of solar radiance are represented by solid lines. The propagation paths of self-emitted thermal radiance are represented by dashed lines. Each path in the figure is numbered at the sensor and corresponds to the radiance values L_1 to L_{12} , each with the units of ($W/(m^2 \cdot sr)$).

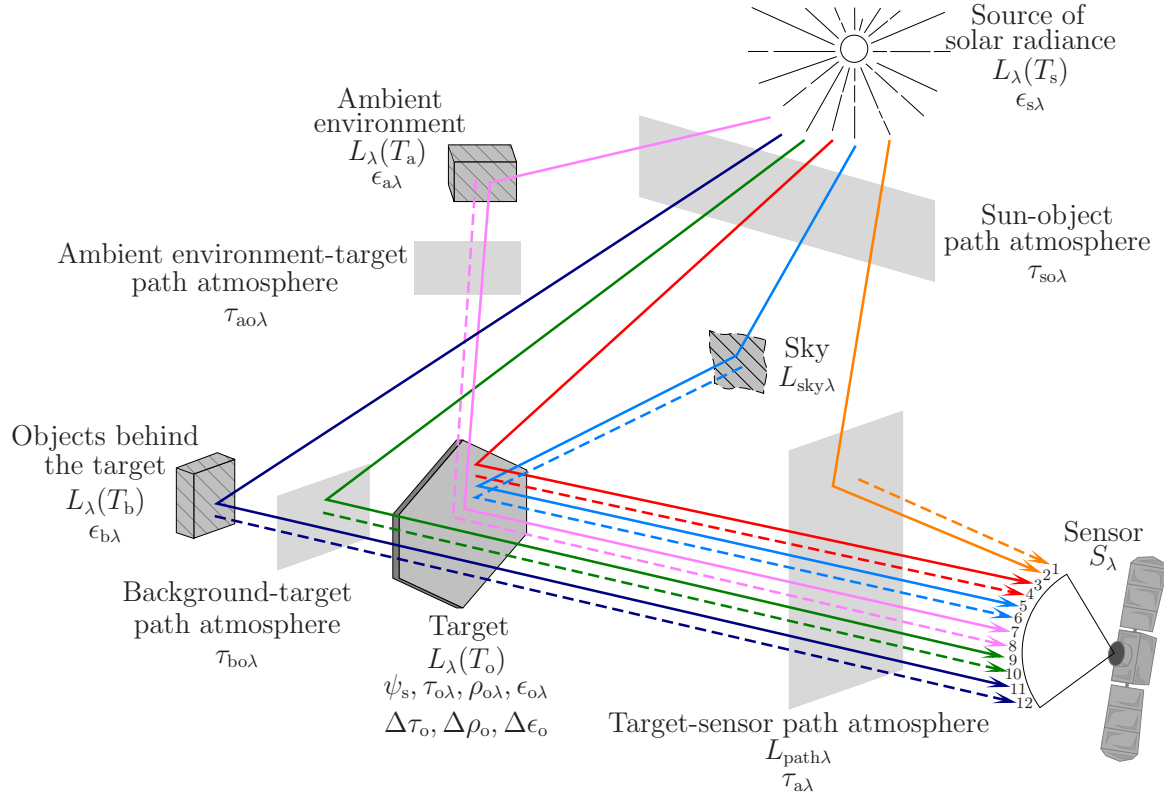


Figure 4.1: Illustration of the OSSIM signature rendering equation. The solar radiance propagation paths are represented by solid lines. The self-emitted thermal radiance propagation paths are represented by dashed lines. (Image adapted from [6] and [9, p. 280]).

Equations (4.2.2) to (4.2.8) relate the radiance propagation paths, L_1 to L_{12} , in Figure 4.1, to the terms in (4.2.1).

Reflected sunlight component: This term describes the surface-reflected radiance [2, p. 47]. The reflected sunlight allows the variations in the surface reflectance of an object to be modelled in terms of diffuse and specular reflection. The variations in the reflectance provide characteristic information about the object [1, p. 58]. The reflected solar radiance depends on the BRDF properties of the object, which is modelled as $f_o(\theta_s, \theta_v, \varphi_s, \varphi_v)$ and has the unit (sr^{-1}). In the OSSIM environment, the very simple Phong phenomenological BRDF model is used to provide a basic BRDF capability, but developers can add their own model if required [39, p. 8.3.6]. The reflected sunlight radiance component is defined as,

$$L_3 = \Delta\rho_o \psi_s \cos\theta_s \int_{\lambda_0}^{\lambda_1} \epsilon_{s\lambda} L_\lambda(T_s) \tau_{so\lambda} f_o(\theta_s, \theta_v, \varphi_s, \varphi_v) \tau_{a\lambda} S_\lambda d\lambda. \quad (4.2.2)$$

where ψ_s is calculated using the area of the sun (A_{sun}) and the distance between the object and the sun (d_{sun}) [9, p. 282],

$$\psi_s = \frac{A_{\text{sun}}}{d_{\text{sun}}^2 \pi}. \quad (4.2.3)$$

In this study the object being considered is on the surface of the earth. Therefore, d_{sun} has a mean value of 149.6×10^6 km. The sun area is defined as 1.523×10^{12} km². The value of ψ_s is then calculated as being 2.17×10^{-5} and is unitless.

Self-emitted thermal radiance component: The self-emitted thermal radiance of a target provides information about the target's temperature and emissivity [1, p. 61]. The radiated flux depends on the directional emissivity ($\epsilon_{o\lambda}(\theta_v, \varphi_v)$) of the target [9, p. 281]. The self-emitted thermal radiance component is defined as,

$$L_4 = \Delta\epsilon_o \int_{\lambda_0}^{\lambda_1} \epsilon_{o\lambda}(\theta_v, \varphi_v) L_{\lambda}(T_o) \tau_{a\lambda} S_{\lambda} d\lambda. \quad (4.2.4)$$

Atmospheric path radiance component: The atmosphere between the target and the sensor adds additional radiance to the total radiance received by the sensor [2, p. 49], [9, p. 283]. The path radiance does not carry any target information content [18, p. 7]. This radiance causes the contrast of images to become washed out and is a function of how 'hazy' or hot the atmosphere is [1, p. 58], [9, p. 118]. The scattered sunlight path radiance and self-emitted thermal path radiance components are collectively referred to as the upwelled radiance [1, p. 62]. MODTRAN or similar RTC models can be used to calculate the path radiance [7, p. 117]. The path radiance component is defined as,

$$L_1 + L_2 + L_9 + L_{10} = \int_{\lambda_0}^{\lambda_1} L_{\text{path}\lambda} S_{\lambda} d\lambda. \quad (4.2.5)$$

Diffuse reflected sky component: Skylight is a diffuse radiance source, caused by atmospheric scattering and emission [9, p. 283]. Skylight increases as the atmosphere becomes more hazy, overcast or hot [1, p. 58]. The solar scattered and thermal self-emitted skylight components are collectively referred to as the downwelled radiance [1, p. 61]. Skylight is difficult to model accurately and is usually measured or calculated using an RTC [9, p. 283]. The reflected skylight radiance component is defined as,

$$L_5 + L_6 = \Delta\rho_o \cos\theta_a \int_{\lambda_0}^{\lambda_1} \int_{\text{sky}} L_{\text{sky}\lambda} \rho_{o\lambda} \tau_{a\lambda} S_{\lambda} d\Omega d\lambda. \quad (4.2.6)$$

Transmitted background radiance component: This term accounts for the effect of semi-transparent targets in a scene. Flux emitted from objects placed behind a semi-transparent target is transmitted through the target towards the sensor [9, p. 283]. If the target is opaque, then this term has a zero contribution to the at-sensor radiance [9, p. 283]. The transmitted background radiance component is defined as,

$$L_{11} + L_{12} = \Delta\tau_o \int_{\lambda_0}^{\lambda_1} \epsilon_{b\lambda} L_{\lambda}(T_b) \tau_{bo\lambda} \tau_{o\lambda} \tau_{a\lambda} S_{\lambda} d\lambda. \quad (4.2.7)$$

Diffuse reflected ambient environment component: This term describes the reflected and self-emitted radiance from surrounding objects and the terrain, that is incident onto the target [1, pp. 58 and 62], [9, p. 283]. This term accounts for the trapping effect which describes the ambient environment radiance that bounces a number of times between the target and atmosphere before being collected by the sensor [1, p. 61], [18, p. 9]. It is assumed that the radiance is incident on the target from all directions [9, p. 283]. The reflected ambient environment radiance component is defined as,

$$L_7 + L_8 = \Delta\rho_o \int_{\lambda_0}^{\lambda_1} \int_{\text{amb}} \epsilon_{a\lambda} L_\lambda(T_a) \tau_{ao\lambda} \rho_{o\lambda} \tau_{a\lambda} S_\lambda \, d\Omega \, d\lambda. \quad (4.2.8)$$

The OSSIM ‘skydome’ functionality can also be used to model the trapping effect for radiance bouncing between the atmosphere and target. The skydome, however, cannot model the trapping effect where radiance bounces between vertical structures, such as buildings; this is set as future work. The skydome functionality models radiance of the sky or environment along any direction over a full sphere, as well as the hemispherical irradiance on a surface pointing along any direction over a full sphere [7, p. 60]. The skydome includes the sky (upper hemisphere) and the terrain (lower hemisphere) [7, p. 60]. The skydome radiance function uses MODTRAN to calculate the path or terrain radiance for a slant path to space or to the ground [7, p. 60]. The skydome radiance quantity combines the sky radiance term (4.2.6) and the reflected ambient radiance term (4.2.8) in a single data structure. The terrain is modelled only by a spectral albedo [7, p. 60].

4.3 Including the adjacency effect in the signature rendering equation

The OSSIM signature rendering equation is expanded to include the adjacency effect when modelling the satellite at-sensor radiance. The adjacency effect occurs when the flux from objects adjacent to the target is scattered by atmospheric aerosol particles into the sensor FOV, towards the target, as upward path radiance [1, p. 59], [2, p. 49], [7, p. 117], [18, p. 8]. The flux from adjacent pixels mixes together with the target pixel flux, such that the two flux sources cannot be discriminated in the image [19], [20]. For this study the adjacency effect is classified as either ‘out-of-FOV’ adjacency or ‘in-FOV’ adjacency. The out-of-FOV adjacency effect occurs when the radiance of objects, or pixels, far outside the sensor FOV is scattered into the target pixel instantaneous field of view (IFOV) as upward path radiance [1, p. 61], [7, p. 118], [40]. The in-FOV adjacency effect occurs when the radiance of pixels inside and near the sensor FOV is scattered into the target pixel IFOV as upward path radiance [2, p. 49], [7, p. 117], [17]. This study considers only the in-FOV adjacency effect. Figure 4.2 illustrates the in-FOV and out-of-FOV adjacency effects. For simplicity and illustration purposes, only two pixels are shown in the sensor

FOV. Each path in the figure is numbered at the sensor and corresponds to the radiance values L_{13} to L_{16} , each with the units of $(W/(m^2 \cdot sr))$.

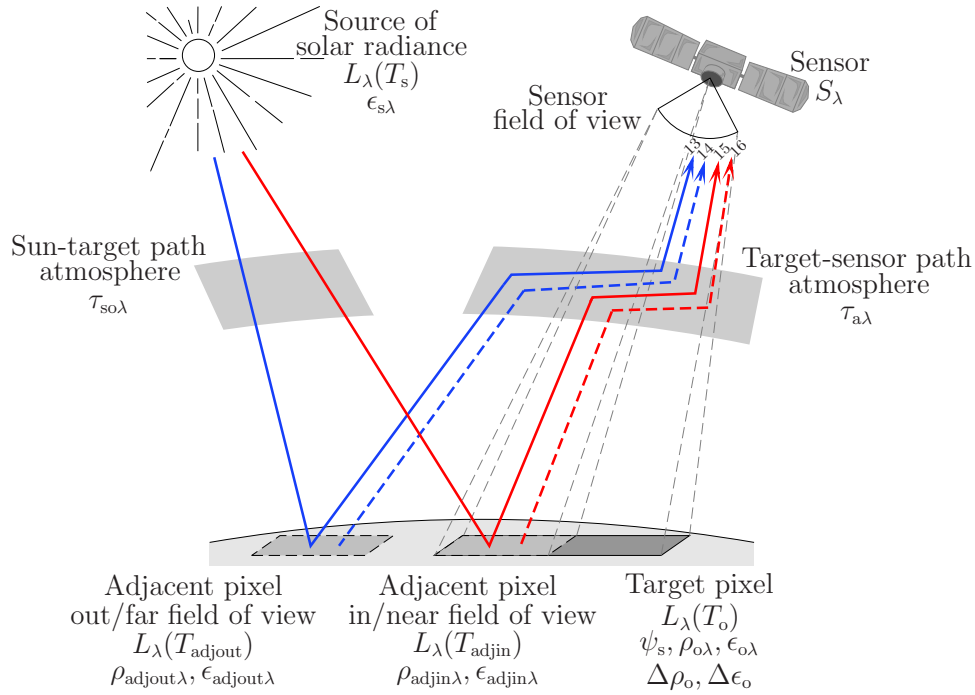


Figure 4.2: The in-field-of-view and out-of-field-of-view adjacent pixel radiance. The solar radiance propagation paths are represented by solid lines. The self-emitted thermal radiance propagation paths are represented by dashed lines.

Equations (4.3.1) and (4.3.2) describe the radiance propagation paths, L_{13} to L_{16} , depicted in figure 4.2.

Out-of-FOV adjacency effect component: This term describes the reflected and self-emitted radiance from objects far outside of the sensor FOV that is scattered into the target pixel IFOV to contribute to the at-sensor radiance. The out-of-FOV adjacency component is represented by,

$$\begin{aligned} L_{13} + L_{14} &= \int_{\lambda_0}^{\lambda_1} \epsilon_{\text{adjout}\lambda} L_{\lambda}(T_{\text{adjout}}) \tau_{a\lambda} S_{\lambda} d\lambda \\ &= \int_{\lambda_0}^{\lambda_1} L_{\text{outFovAdj}\lambda} S_{\lambda} d\lambda. \end{aligned} \quad (4.3.1)$$

Equation (4.3.1) is not mathematically rigorous and serves only to define the various contributors to the out-of-FOV adjacency radiance. $L_{\lambda}(T_{\text{adjout}})$ represents the spectral black body radiance at the out-of-FOV object at a temperature of T_{adjout} . The spectral emissivity of this object is represented by $\epsilon_{\text{adjout}\lambda}$. The $L_{\text{outFovAdj}\lambda}$ variable is then used to represent these out-of-FOV radiance contributing factors for all objects in the far-FOV.

To model the out-of-FOV adjacency effect a very large high resolution image is required. Due to computational constraints, it was not possible to generate a large enough high

resolution scene image that includes far-FOV pixels. For this study the out-of-FOV adjacency effect was not considered and the implementation of out-of-FOV adjacency effect has been set as future work.

In-FOV adjacency effect component: This term describes the reflected and self-emitted radiance from objects inside or near to the sensor FOV that is scattered into the target pixel IFOV, to contribute to the at-sensor radiance. The near-FOV is defined as the area outside the sensor FOV that is still within the bounds of the high resolution image. The in-FOV adjacency component is represented by,

$$\begin{aligned} L_{15} + L_{16} &= \int_{\lambda_0}^{\lambda_1} \epsilon_{\text{adjin}\lambda} L_{\lambda}(T_{\text{adjin}}) \tau_{\text{a}\lambda} S_{\lambda} \, d\lambda \\ &= \int_{\lambda_0}^{\lambda_1} L_{\text{inFovAdj}\lambda} S_{\lambda} \, d\lambda. \end{aligned} \quad (4.3.2)$$

Equation (4.3.2) is not mathematically rigorous and serves only to define various contributors to the in-FOV adjacency radiance. $L_{\lambda}(T_{\text{adjin}})$ represents the spectral black body radiance at the pixel in or near the sensor FOV, at a temperature of T_{adjin} . The spectral emissivity of this object is represented by $\epsilon_{\text{adjin}\lambda}$. The $L_{\text{inFovAdj}\lambda}$ variable is then used to represent these in-FOV radiance contributing factors, for all objects in and near the sensor FOV.

The OSSIM equation, expanded with the in-FOV radiance term, is then given as,

$$\begin{aligned} L_{\Delta\lambda} &= \overbrace{\Delta\rho_o \psi_s \cos\theta_s \int_{\lambda_0}^{\lambda_1} \epsilon_{\text{s}\lambda} L_{\lambda}(T_s) \tau_{\text{so}\lambda} f_o(\theta_s, \theta_v, \varphi_s, \varphi_v) \tau_{\text{a}\lambda} S_{\lambda} \, d\lambda}^{\text{reflected sunlight}} \\ &+ \overbrace{\Delta\epsilon_o \int_{\lambda_0}^{\lambda_1} \epsilon_{\text{o}\lambda}(\theta_v, \varphi_v) L_{\lambda}(T_o) \tau_{\text{a}\lambda} S_{\lambda} \, d\lambda}^{\text{self-emitted thermal radiance}} + \overbrace{\int_{\lambda_0}^{\lambda_1} L_{\text{path}\lambda} S_{\lambda} \, d\lambda}^{\text{atmospheric path radiance}} \\ &+ \overbrace{\Delta\rho_o \cos\theta_a \int_{\lambda_0}^{\lambda_1} \int_{\text{sky}} L_{\text{sky}\lambda} \rho_{\text{o}\lambda} \tau_{\text{a}\lambda} S_{\lambda} \, d\Omega \, d\lambda}^{\text{diffuse reflected sky}} \\ &+ \overbrace{\Delta\tau_o \int_{\lambda_0}^{\lambda_1} \epsilon_{\text{b}\lambda} L_{\lambda}(T_b) \tau_{\text{bo}\lambda} \tau_{\text{o}\lambda} \tau_{\text{a}\lambda} S_{\lambda} \, d\lambda}^{\text{transmitted background}} \\ &+ \overbrace{\Delta\rho_o \int_{\lambda_0}^{\lambda_1} \int_{\text{amb}} \epsilon_{\text{a}\lambda} L_{\lambda}(T_a) \tau_{\text{ao}\lambda} \rho_{\text{o}\lambda} \tau_{\text{a}\lambda} S_{\lambda} \, d\Omega \, d\lambda}^{\text{diffuse reflected ambient environment}} \\ &+ \overbrace{\int_{\lambda_0}^{\lambda_1} L_{\text{inFovAdj}\lambda} S_{\lambda} \, d\lambda}^{\text{in-field-of-view adjacent radiance}}. \end{aligned} \quad (4.3.3)$$

4.4 Implementing the signature rendering equation

The OSSIM signature rendering equation and the additional in-FOV adjacency effect term are implemented to model the total at-sensor radiance for a satellite sensor.

The in-FOV adjacency effect can be modelled by an atmospheric aerosol scattering PSF which is convolved with the high resolution image. The aerosol scattering PSF is discussed in section 5.3.4. This PSF, however, is found to be wider than the number of detectors in the sensor CCD arrays, as well as beyond the size limits of an image that can be computed within a reasonable simulation turn-around time. This wide PSF makes it impractical to implement the in-FOV adjacency as a convolution-computed PSF. The in-FOV adjacency effect can, however, be approximated with a constant flux value over the sensor FOV. This is possible because the intensity of the spatially wide scattering PSF appears approximately constant over the much smaller scene in the sensor FOV. This results in all pixels surrounding any target pixel having an equal contribution to the in-FOV adjacency effect for that target pixel. The constant in-FOV adjacency value is calculated as the average PSF intensity, excluding the centre peak, over the PSF area that corresponds to the high resolution image size. The aerosol scattering PSF can then be thought of as a PSF that is the size of the high resolution image where every pixel, except the central target pixel, has the same value. The sum of all the pixels in this PSF is normalised to ensure that no gain is introduced.

The in-FOV adjacency effect radiance term, using the constant in-FOV adjacency value over the high resolution image size, is expressed as,

$$L_{\text{inFovAdj}} = (C_{\text{const}} \sum_{i=0}^{N_{\text{pixel}}} L_i) - (C_{\text{const}} L_{\text{Ptarget}}) + (C_{\text{peak}} L_{\text{Ptarget}}), \quad (4.4.1)$$

where C_{const} is the constant in-FOV adjacency value over the scene and C_{peak} is the PSF peak value at the target pixel. N_{pixel} is the number of pixels in the high resolution image, L_i is the i^{th} high resolution image pixel radiance value and L_{Ptarget} is the pixel radiance for the target pixel.

The in-FOV adjacency effect is implemented in the OSPA C++ code. The implementation of the in-FOV adjacency effect is summarised using pseudo code, as given in Listing 4.1.

The extent of the in-FOV adjacency effect on the target pixel is limited by the size of the high resolution image.

4.5 Conclusion

A signature rendering equation models the total at-sensor radiance and the effect of the sensor's response on the collected radiance. Each term in the signature rendering equation


```

//-----
//Calculate the in-FOV adjacency effect for each of the pixels in the sensor.
//The number of pixels in and near the sensor FOV that contribute to the
//adjacency effect is dependent on the number of pixels in the scene image
//(high resolution image).
//-----

//--- Add up all the pixel values in the high resolution image plane,
//    for the sensor image band.
sumOfHiresPlane = hiresImage.GetSum(imageBand);

//--- Multiply the sum with the constant in-FOV scattering value.
hiresSumWithConstant = sumOfHiresPlane * ConstValue;

for(row=0; row<number of detector rows in the sensor array; row++)
{
    for(col=0; col<number of detector columns in the sensor array; col++)
    {
        //--- Calculate the in-FOV adjacency effect for the target pixel at the
        //    high resolution image position (pixelRow, pixelCol, imageBand).
        targetPixelAdj = hiresSumWithConstant
            - (hiresImage(pixelRow, pixelCol, imageBand)*ConstValue)
            + (hiresImage(pixelRow, pixelCol, imageBand)*CentrePeakValue);
    } //End loop over columns
} //End loop over rows

```

Listing 4.1: C++ code used to calculate the in-field-of-view adjacency effect term.

describes a different contributor to the signature as well as various factors, such as the atmosphere, that affect the at-sensor radiance.

The OSSIM signature rendering equation is shown to include the necessary terms, with the exception of the adjacency effect term, that are required to model the at-sensor radiance for a satellite sensor. As a result, an expanded version of the OSSIM signature rendering equation is used to model the satellite at-sensor radiance.

For this study the adjacency effect is divided into an in-FOV term and an out-of-FOV term. The in-FOV adjacency effect term can be included in the form of an aerosol scattering PSF. The aerosol scattering PSF is, however, found to be wider than the sensor CCD array and the simulated high resolution scene. The intensity of the spatially wide scattering PSF appears approximately constant over the much smaller scene in the sensor FOV. This results in all the pixels in the simulated scene contributing equally to the in-FOV adjacency radiance at the target pixel. The in-FOV adjacency effect is, therefore, modelled as a constant value over the sensor FOV. Due to computational constraints, the out-of-FOV adjacency effect is not considered in this study.

Chapter 5 provides a discussion on the atmosphere and the aerosol scattering PSF implemented in the simulation.

Chapter 5

The Atmosphere

5.1 Introduction

The atmosphere is an important link in the image chain, since all flux reaching a satellite sensor travels some distance through it. The atmosphere significantly affects the radiance and spectral distribution of the propagating flux [3, pp. 9 and 46], [20]. The atmosphere, therefore, alters the spectral signature of the objects being observed by the satellite sensor [10]. It is important to model the atmospheric effects when calculating the at-sensor radiance. The atmospheric model should take into account the atmospheric radiation, molecular absorption, aerosol scattering and the effects of turbulence [3, pp. 9 and 23], [11, p. 444], [20], [41].

Atmospheric effects are included in the simulation environment by means of an RTC. The atmospheric radiation is accounted for in the signature rendering equation (4.3.3), in section 4.3, in the form of atmospheric path radiance and reflected sky radiance. Atmospheric aerosol scattering causes the adjacency effect to occur. The adjacency effect can be modelled by means of an aerosol scattering PSF discussed in section 4.3. Atmospheric turbulence is beyond the scope of this study and was not considered.

The total effect of the atmosphere on the propagating radiance varies with the path length, atmospheric conditions, time of day, location and wavelength [3, pp. 9 and 46]. In this study the top of the atmosphere was set to be at 100 km above sea level, beyond which no atmospheric effects are modelled.

5.2 Radiative transfer codes

RTCs are an important part of a satellite remote sensing tool suite [12]. RTCs are used to model the atmosphere by calculating the atmospheric scattering and absorption effects [13]. RTCs are also used when estimating the top-of-atmosphere (TOA) spectral radiance [12], [13]. The need for reliable and straightforward ways to perform radiative transfer

calculations has led to numerous RTCs being developed. RTCs differ in their ability to model different atmospheric conditions [42].

Several RTCs are available to simulate the atmosphere. Examples of some RTCs that are used in satellite data processing algorithms are the Second Simulation of a Satellite Signal in the Solar Spectrum Vector code (6SV), MODTRAN, Radiative Transfer 3 (RT3) and SHARM [12]. 6SV and MODTRAN are the RTCs most commonly used by the Committee on Earth Observation Satellite-Working Group on Calibration and Validation [13].

MODTRAN models atmospheric conditions over a wider spectral range than 6SV [18, p. 32], [43, p. 21]. 6SV calculates the radiance measured by a satellite over the visible, NIR and MWIR spectral ranges (0.25 μm to 4.0 μm) [18, p. 32]. MODTRAN, however, calculates the radiance over the broader spectral range of 0.2 μm to 200 μm [43, p. 21]. 6SV accounts for radiation polarisation, whereas MODTRAN does not [13], [18, p. 3]. Radiation polarisation, however, was not considered in this study.

MODTRAN is one of the most widely used RTCs and is recommended for use in remote sensing applications [13], [41]. MODTRAN is well tested and allows the user to evaluate the effects of many atmospheric and surface reflectance characteristics on the TOA radiance [42]. MODTRAN models atmospheric radiative transfer effects for a number of different viewing geometries and includes multiple scattering and different aerosol specification schemes [42], [43, pp. 12 and 25].

MODTRAN4 was a suitable RTC to use for this study, was available to the author and was integrated in OSSIM.

5.3 Description of the implemented atmosphere

The atmospheric transmittance, path radiance, optical depth and aerosol scattering PSF are required when simulating the atmosphere in the OSPA environment. OSPA uses the MODTRAN calculated atmospheric transmittance and path radiance in the signature rendering equation. The aerosol scattering PSF and MODTRAN calculated optical depth are used when calculating the in-FOV adjacency effect.

5.3.1 MODTRAN atmospheric calculation settings

MODTRAN calculates the atmospheric values for a specific location, date and time. For this study, an atmospheric model was constructed for a site near Pretoria (25°29'45"S, 28°22'25"E). This location corresponds to the Paardefontein calibration site, as described in Appendix C.1.3. The ground altitude for this location is 1.2 km above sea level. The WorldView-2 satellite overpass time and date for this location was not available to the author; as a result an overpass time was calculated and a date was

assumed. The overpass time was calculated using the WorldView-2 satellite 10:30 a.m equatorial crossing time, 770 km orbit altitude, 97.8° orbit inclination angle and the satellite platform velocity [44].

The platform velocity (v) for WorldView-2 is calculated to be 7471.89 m/s by using

$$v = \sqrt{(g r_{\text{EARTH}}^2)/(r_{\text{EARTH}} + H)}, \quad (5.3.1)$$

where gravity (g) is equal to 9.81 m/s^2 , the radius of the earth (r_{EARTH}) is 6378.1 km and H is the orbital altitude (m).

The distance the satellite will travel from the equator to the Pretoria overpass location is calculated using spherical trigonometric identities [45],

$$\cos(k) = \cos(l) \cos(m) + \sin(l) \sin(m) \cos(K), \quad (5.3.2a)$$

$$\sin(k) \cos(L) = \cos(l) \sin(m) - \sin(l) \cos(m) \cos(K). \quad (5.3.2b)$$

The geometry of (5.3.2) is illustrated in Figure 5.1.

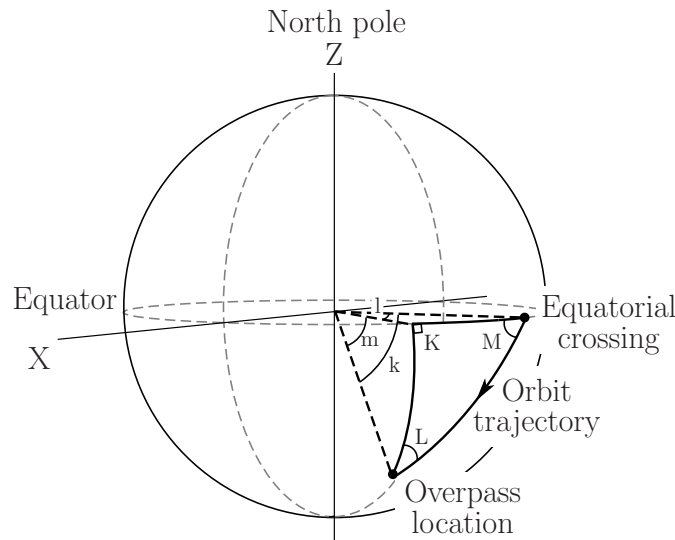


Figure 5.1: Illustration of the angles of a triangle on a sphere. The triangle points are at the overpass location longitude equatorial crossing (at angle K), the satellite overpass location (at angle L) and the satellite orbit equatorial crossing (at angle M).

The angle K is between the equator and overpass location longitude and has a value of 90° . Angle k is the arc angle for the orbit trajectory path between the equator and overpass location. Angle L is the angle between the overpass location longitude and the orbit trajectory. Using the orbit inclination angle, the angle L was calculated to be $97.8^\circ - 90.0^\circ = 7.8^\circ$, which is 0.136 rad. Angle l is the arc angle for the path between the overpass location longitude and the equator crossing longitude. Angle M is the angle between the equator and the orbit trajectory. Angle m is the arc angle, along the overpass location longitude, between the equator and the overpass location and has a value equal to the overpass location latitude of 25.49583° , which is 0.445 rad.

The angle values are substituted into (5.3.2) which results in

$$\cos(k) = \cos(l) \cos(0.445)$$

$$\therefore \cos(l) = \frac{\cos(k)}{0.903}, \quad (5.3.3a)$$

and

$$\sin(k) \cos(0.136) = \cos(l) \sin(0.445)$$

$$\therefore \sin(k) = \frac{0.430 \cos(l)}{0.991}. \quad (5.3.3b)$$

Equation (5.3.3a) is then substituted into (5.3.3b) and k (rad) is solved as

$$\sin(k) = \frac{0.430(\cos(k)/0.903)}{0.991}$$

$$\therefore \tan(k) = \frac{0.430}{0.991 \times 0.903}$$

$$\therefore k = 0.448. \quad (5.3.4)$$

Using the calculated value of k , the arc distance (s) between the equator and location is then calculated to be 3206.696 km by using

$$s = (r_{\text{EARTH}} + H)k, \quad (5.3.5)$$

The time t taken for WorldView-2 to travel the distance s is then calculated to be 7.15 min by using

$$t = \frac{s}{v}. \quad (5.3.6)$$

The descending overpass time at the Pretoria location is then calculated to the nearest minute to be

$$10:30 \text{ a.m.} + 7 \text{ min} = 10:37 \text{ a.m.} \quad (5.3.7)$$

The Pretoria atmosphere in March is similar to the MODTRAN Tropical atmospheric model (27 °C, 75% relative humidity at sea level); as a result, the atmosphere is calculated for day 60 (2014-03-01) using this MODTRAN model.

It is desirable to have a clear atmosphere when taking satellite images. For this study the atmosphere is assumed to be cloud free and have good visibility conditions. To simulate good visibility through the atmosphere, a tropospheric aerosol model with a visibility of 50 km is used. The atmospheric carbon dioxide concentration is set to 380 ppmv.

Since the satellite is viewing the target at nadir, the atmospheric path properties are calculated for a zenith angle of 180°. The MODTRAN atmospheric model extends to an altitude of 100 km above sea level.

For this study the targets are assumed to be surrounded by dry grass, which matches the vegetation in the Paardefontein area. The atmospheric model is therefore set to use the MODTRAN 'dry grass' albedo model at the start of the atmospheric path.

The atmospheric values are calculated for a slant path to the ground.

5.3.2 Atmospheric transmittance

Atmospheric transmittance is defined as the ratio of the source flux reaching the sensor with the atmosphere present, to that reaching it when no atmosphere is present [9, p. 98]. For comparison purposes the atmospheric transmittance for both a good visibility atmosphere and for a bad visibility atmosphere, calculated using MODTRAN atmospheric profiles, are plotted in Figure 5.2. Poor visibility conditions are simulated using the MODTRAN rural aerosol model with a visibility of 5 km. Good visibility conditions are simulated using the MODTRAN tropospheric aerosol model with a visibility of 50 km.

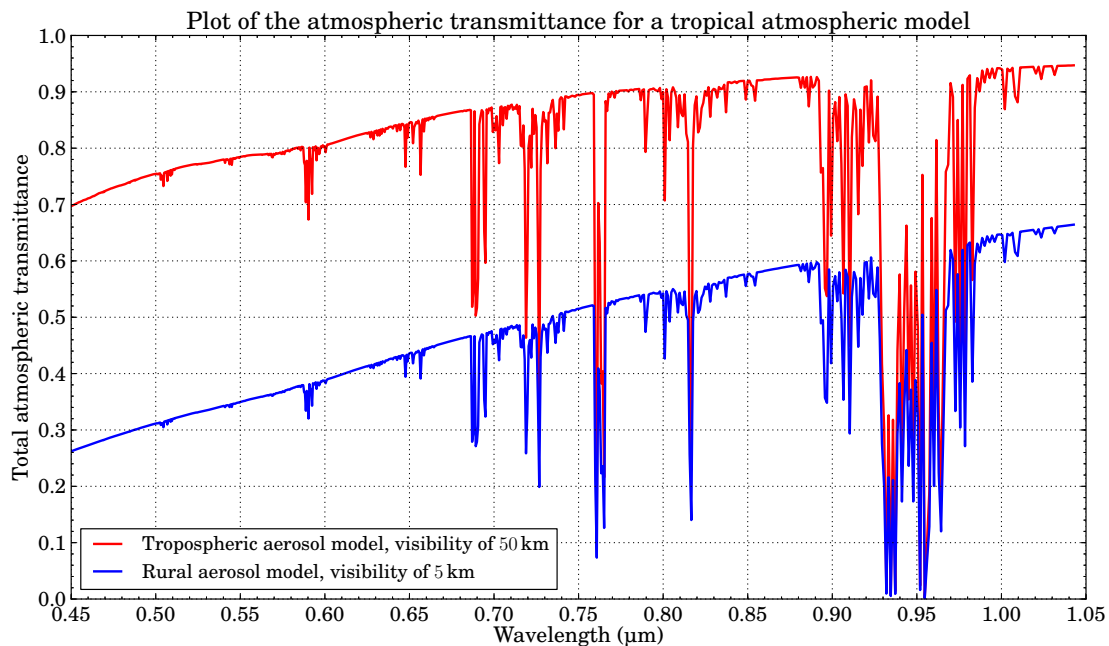


Figure 5.2: MODTRAN atmospheric transmittance for a Tropical atmospheric model with different visibilities.

From Figure 5.2 it is evident that a portion of the propagating radiance is attenuated (absorbed or scattered) by the atmosphere. Under good visibility conditions of 50 km, 70% to 95% of the propagating radiance in the visible and NIR range is transmitted through the atmosphere. Molecular scattering dominates in this model, while the aerosol scattering is negligible. Under poor visibility conditions of 5 km, only 26% to 66% of the propagating radiance in the visible and NIR range is transmitted through the atmosphere. Rural aerosol scattering dominates in this model.

5.3.3 Atmospheric optical depth

The optical depth, also known as optical thickness, is a measure of the opacity of a medium [9, p. 103]. Figure 5.3 plots the atmospheric optical depth for the two MODTRAN atmospheric profiles used to plot the atmospheric transmittance in Figure 5.2.

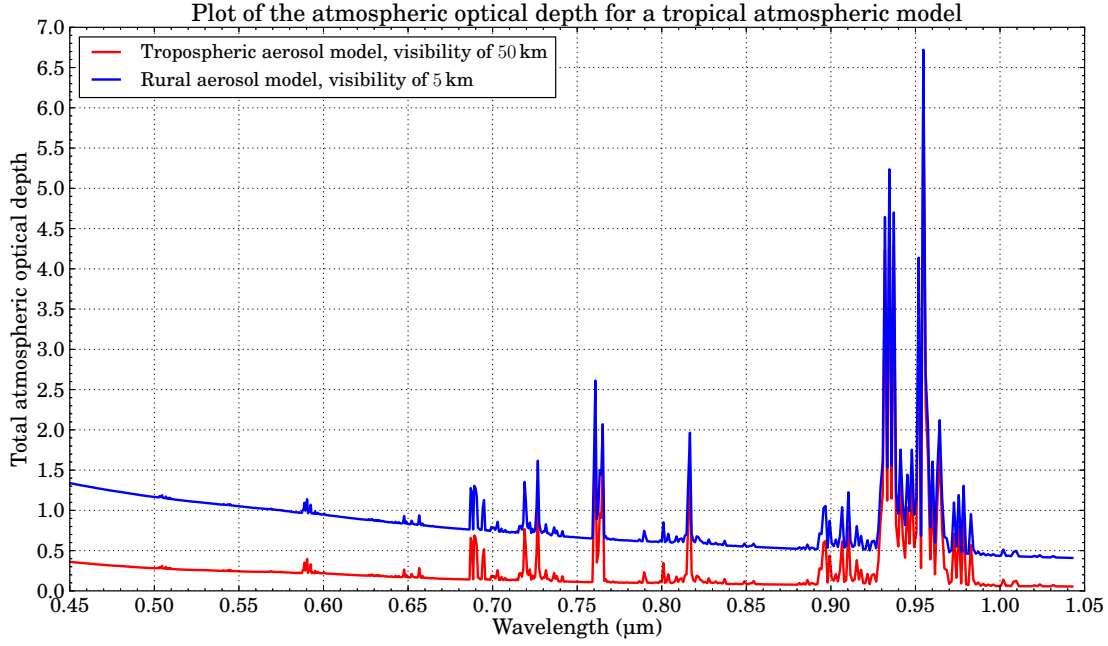


Figure 5.3: MODTRAN atmospheric optical depth for a Tropical model with different visibilities.

As seen in Figure 5.3, under good visibility conditions of 50 km, the optical depth values are low, below 0.4, in the visible and NIR atmospheric transmittance windows. Under poor visibility conditions of 5 km, it is seen that the optical depth values are higher, between 0.4 and 1.4, in the visible and NIR atmospheric transmittance windows. This indicates that the atmosphere is more opaque under conditions of poor visibility.

From Figures 5.2 and 5.3 it is evident that atmospheric transmittance and optical depth have an inverse relationship [9, p. 103].

5.3.4 Atmospheric aerosol scattering point spread function

The atmospheric aerosol scattering PSF is calculated from the atmospheric aerosol scattering modulation transfer function (MTF).

The atmospheric aerosol MTF (unitless) is given as [19], [20]

$$MTF_a(f_r) = \begin{cases} \exp \left\{ - \left(\frac{f_r}{f_c} \right)^2 \sec \theta \int_{h_t}^H [S_a(h, \lambda) + A_a(h, \lambda)] dh \right\}, & \text{if } f_r < f_c \\ \exp \left\{ - \sec \theta \int_{h_t}^H [S_a(h, \lambda) + A_a(h, \lambda)] dh \right\}, & \text{if } f_r \geq f_c. \end{cases} \quad (5.3.8)$$

where λ is the imaging wavelength (m), f_r is the spatial frequency and f_c is the high spatial frequency cut-off, both of which are expressed as cycles/rad or cycles/m. The spatial frequency cut-off is calculated as

$$f_c = \frac{r_a}{\lambda f}, \quad (5.3.9)$$

where r_a is the radius (m) of the scattering particle and f' is the imaging system focal length (m) [19], [20]. Equation (5.3.9) expresses f_c as cycles/m in the focal plane. The spatial cut-off frequency is expressed in cycles/rad by removing the projection onto the focal plane,

$$f_c = \left(\frac{r_a}{\lambda f'} \times f' \right) = \frac{r_a}{\lambda}. \quad (5.3.10)$$

In (5.3.8), $S_a(h, \lambda)$ and $A_a(h, \lambda)$ are the spectral atmospheric attenuation scattering and absorption coefficients respectively, with units of (m^{-1}). h_t is the target altitude (m) above the ground and θ is the viewing angle (rad) relative to a vertical path [19], [20].

The optical depth ($\sigma_{\text{od}}(h, \lambda)$) is unitless and is calculated as [1, pp. 97 and 113]

$$\sigma_{\text{od}}(h, \lambda) = \sec \theta \int_{h_1}^H [S_a(h, \lambda) + A_a(h, \lambda)] dh. \quad (5.3.11)$$

Equation (5.3.8) is rewritten in terms of the optical depth as

$$MTF_a(f_r) = \begin{cases} \exp \left[- \left(\frac{f_r}{f_c} \right)^2 \sigma_{\text{od}}(h, \lambda) \right], & \text{if } f_r < f_c \\ \exp [-\sigma_{\text{od}}(h, \lambda)], & \text{if } f_r \geq f_c. \end{cases} \quad (5.3.12)$$

Figure 5.4 shows a plot of the atmospheric aerosol MTF, using (5.3.12), for the WorldView-2 satellite panchromatic band with a peak wavelength of 776 nm. The cut-off frequency is calculated using (5.3.10) to be 0.644 cycles/rad for an aerosol particle radius of 0.5 μm . The satellite viewing angle is set to zero for at-nadir imaging. The optical depth is for a MODTRAN tropical atmosphere with a 50 km visibility tropospheric aerosol model. The optical depth value is calculated, using MODTRAN, to be 0.103.

The atmospheric aerosol PSF is calculated by taking the inverse Fourier transform of the atmospheric aerosol MTF, resulting in

$$PSF_a(\omega_r) = \begin{cases} \sqrt{\frac{\pi}{x}} \exp \left(-\frac{\pi^2 \omega_r^2}{x} \right), & \text{if } |\omega_r| > \frac{\omega_{\text{pixelPitch}}}{2} \\ \exp [-\sigma_{\text{od}}(h, \lambda)] \delta(\omega_r), & \text{if } |\omega_r| \leq \frac{\omega_{\text{pixelPitch}}}{2}, \end{cases} \quad (5.3.13)$$

where ω_r is the angular frequency (rad) over which the PSF is calculated and $\omega_{\text{pixelPitch}}$ is the angular frequency (rad) for a single pixel. The variable x , in (5.3.13), is calculated using

$$x = \frac{\sigma_{\text{od}}(h, \lambda)}{f_c^2} = \sigma_{\text{od}}(h, \lambda) \omega_c^2, \quad (5.3.14)$$

where ω_c is the angular cut-off frequency (rad).

A detailed derivation of the atmospheric aerosol PSF is provided in Appendix D.

The normalised atmospheric aerosol PSF for the WorldView-2 satellite panchromatic band, with a peak wavelength of 776 nm, is calculated using (5.3.13) and plotted in Figure 5.5.

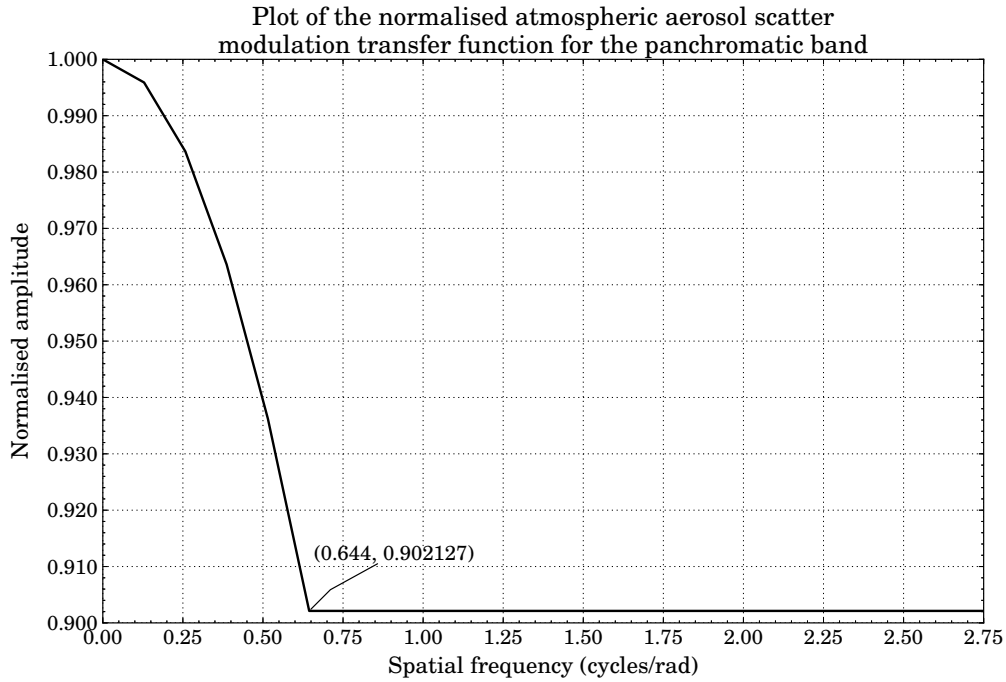


Figure 5.4: Atmospheric aerosol modulation transfer function for at-nadir viewing, wavelength=776 nm, particle radius=0.5 μm , optical depth=0.103.

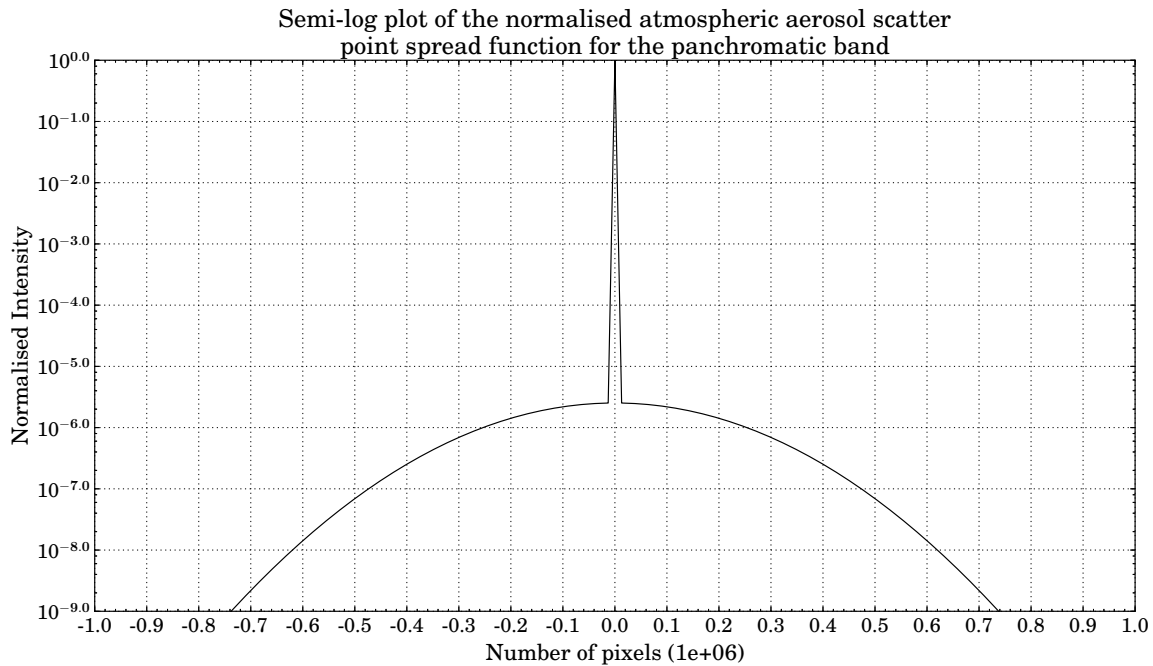


Figure 5.5: Atmospheric aerosol point spread function for at-nadir viewing, wavelength=776 nm, particle radius=0.5 μm , optical depth=0.103.

A fine resolution is required to be able to model the PSF centre spike as being a single pixel wide and also, at the same time, model the PSF at distances further away from the

centre spike. This results in large data sets, which was not possible to generate with the memory available on the computer used in this study. As a result, the PSF centre spike in the spatial domain could not be modelled as one pixel wide, while still achieving the correct modulation transfer function at the cut-off frequency.

For this study the atmospheric aerosol scattering PSF in Figure 5.5 was used to approximate the constant in-FOV adjacency value.

5.4 Conclusion

It is important to model the effects of the atmosphere when calculating the at-sensor radiance. The atmospheric model should take into account the atmospheric radiation, absorption, scattering and the effect of turbulence on the propagating radiance [11, p. 444], [20], [41]. In the OSPA environment the atmospheric aerosol scattering PSF, transmittance, path radiance and optical depth are required when simulating the atmosphere.

MODTRAN is the most widely used RTC and is recommended to be used in remote sensing applications [13], [41]. MODTRAN4 was used to simulate the atmosphere for the satellite remote sensing system model. For this study MODTRAN4 was set to simulate the atmosphere outside the Pretoria area ($25^{\circ}29'45''\text{S}$, $28^{\circ}22'25''\text{E}$) at 10:37 a.m. for day 60 (2014-03-01). The atmosphere is simulated for good visibility conditions.

The next important link in the satellite remote sensing system model is the satellite sensor. Chapter 6 provides a description of the OSPA generic satellite sensor model.

Chapter 6

The Satellite Sensor

6.1 Introduction

An earth observation sensor comprises electronic, optical and mechanical elements to measure and record the radiance of the scene [1, pp. 153 and 167]. Optical elements are used to collect and focus optical flux onto the detector array in the focal plane. Mechanical elements are used to point and steer the optics and in so doing control the line of sight of the detector arrays. The optical detectors convert the optical flux to an electronic signal for further processing. The electronic signal is processed by the sensor electronics in order to prepare the signal for storage or transmission to a ground station [1, p. 167], [2, p. 76].

During the design phase, the sensor designer must make a trade off to balance the spatial, spectral and radiometric resolution requirements of the sensor [1, p. 215]. The precision with which the sensor measures the scene's radiance depends on the spatial and spectral resolution of the sensor [2, pp. 76-77]. The spatial resolution defines the smallest discernible object that can be detected by the sensor [8, p. 39]. The spectral resolution defines the ability of the sensor to resolve detail in a spectral signal [2, p. 82]. The sensor's radiometric resolution defines its ability to detect small differences in the reflected and emitted radiance and is dependent on noise and signal quantisation [1, p. 215], [2, p. 76], [8, p. 43].

The space environment has a significant effect on the design and operation of a satellite and sensor. For instance, the components on the sun side of the satellite operate under conditions of higher temperature than do the components on the shadow side of the satellite [29]. The effect of the space environment on the sensor was not accounted for in this study.

This investigation led to the development of a generic sensor suitable for the modelling of typical satellite sensors. The model included optical, radiometric, detector array and elementary electronics characteristics.

6.2 A generic satellite sensor model

A generic satellite sensor model was designed and implemented that allows a user to simulate different sensor configurations for passive remote sensing. The sensor model accounts for the spatial response, spectral response and radiometric resolution of the detector and optics components. The model also models key radiometric characteristics of a sensor.

The sensor model includes a configurable CCD array model, which allows the user to specify the detector shape, size, response, dwell time, noise and non-uniformity. The sensor model allows the setup of different CCD array layouts for both along-track and across-track scanners and allows for a different detector array structure for each spectral band CCD array. The CCD layout accounts for staggered and shifted detector elements.

The sensor model also includes a telescope optics model. The optics model allows the user to specify the reflective optics spectral response, the telescope optics f-number, focal length and obscuration area.

An ADC process is included in the sensor model. The ADC process produces a DN representation of the measured data. The user must specify the number of bits used for the data quantisation.

The sensor parameters in each of the sensor spectral bands are handled separately, with potentially different sensor configurations. The WorldView-2 satellite sensor configuration is used to set up the generic sensor model. The key sensor model settings are outlined in section 6.3.

6.3 Implemented generic satellite sensor model

6.3.1 Sensor spectral response

Earth observation sensors are designed to operate over a broad spectral range [1, p. 165]. The sensor model spectral range is defined for each of the bands, in terms of wavenumber range limits and a spectral response curve.

The WorldView-2 satellite uses a pushbroom imager that can acquire both panchromatic imagery and multispectral imagery [21], [44], [46]. The panchromatic imagery is recorded over the spectral range of 0.45 μm to 0.8 μm . For multispectral imagery, the sensor uses 8 multispectral bands, which include the standard blue (0.45 μm to 0.51 μm), green (0.51 μm to 0.58 μm), red (0.63 μm to 0.69 μm) and NIR-1 (0.77 μm to 0.895 μm) bands and four additional bands. The additional bands include the coastal blue (0.4 μm to 0.45 μm), yellow (0.585 μm to 0.625 μm), red edge (0.705 μm to 0.745 μm) and NIR-2 (0.860 μm to 1.040 μm) bands [44], [46], [47].

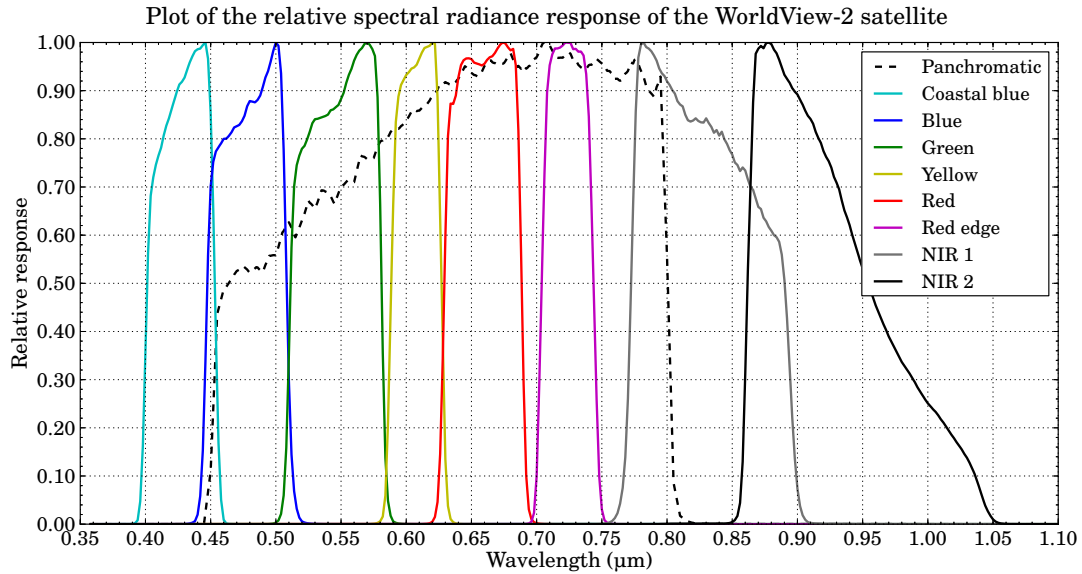


Figure 6.1: WorldView-2 satellite sensor panchromatic and multispectral spectral band definitions (image reproduced from [21]).

The spectral response of each of the WorldView-2 imaging spectral bands is illustrated in Figure 6.1.

6.3.2 Optics response

Earth observation systems make use of reflective optics to focus the collected flux onto the focal plane. The reflective optics consist of a set of mirrors. The WorldView-2 satellite sensor implements an on-axis Korsch type II TMA telescope design [44], [46], [48]. The Korsch type II TMA telescope configuration makes use of four mirrors [22]. Each of the mirrors is coated with FSS99-500 silver metallic coating from Quantum coating [23].

The spectral reflectance curve for a single FSS99-500 silver metallic coating and the telescope's total reflectance is shown in Figure 6.2.

Each mirror in the telescope will reflect a portion of the incident flux onto the next mirror. The more mirrors used in the telescope, the less of the collected flux will reach the detectors, since each mirror will further reduce the propagating flux. The total effective optics reflectance $\rho_{\text{totalMirror}}$ (unitless) is calculated using

$$\rho_{\text{totalMirror}} = \rho_{1\text{mirror}} \times \rho_{2\text{mirror}} \times \dots \times \rho_{(N-1)\text{mirror}} \times \rho_{N\text{mirror}} = \rho_{\text{mirror}}^{N_{\text{mirror}}}, \quad (6.3.1)$$

where ρ_{mirror} is the reflectance of a single mirror and N_{mirror} is the number of reflective mirrors in the telescope. All mirrors are set to have the same reflectance value.

6.3.3 Spatial blurring point spread function kernel

The spatial blurring present in the output image is calculated by a convolution of the sensor image with a PSF kernel. The PSF kernel of the sensor optics, the detectors and

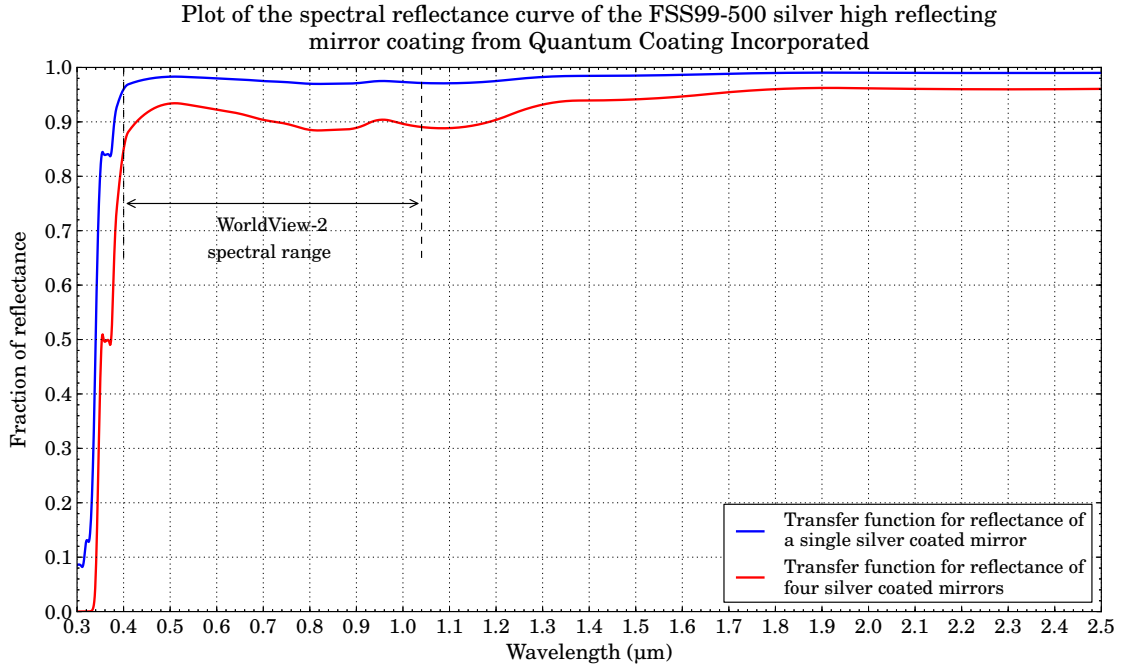


Figure 6.2: Spectral reflectance for the FSS99-500 mirror coating from Quantum Coating Incorporated for a single mirror and the calculated spectral curve for four mirrors [49].

electronics are all convolved to form the total sensor PSF kernel [2, p. 85]. A different normalised PSF kernel is calculated for each of the sensor's spectral bands.

The optics PSF is defined as the spatial energy distribution of the image of a point source in the focal plane. For diffraction-limited circular optics, this distribution is called the Airy Pattern [2, p. 86]. For this study, the optical system is assumed to be diffraction-limited, with a circular aperture and centre obscuration, due to a secondary mirror. The spider arms supporting the secondary mirror are assumed to have a negligible effect on the optics PSF and are, therefore, not considered in this study. The optics PSF blurring kernel for a centrally obscured circular optics is calculated using [50]

$$PSF_{\text{optics}}(x, y) = \frac{1}{(1 - \epsilon^2)^2} \left[\frac{2J_1(\gamma)}{\gamma} - \frac{2\epsilon J_1(\epsilon \gamma)}{\gamma} \right]^2, \quad (6.3.2a)$$

where

$$\gamma = \frac{\pi r D}{\lambda f'}, \quad (6.3.2b)$$

where (x, y) is the location (m) in the image plane, ϵ is the obscuration area ratio (unitless), $J_1(\gamma)$ and $J_1(\epsilon \gamma)$ are first order Bessel functions, $r = \sqrt{x^2 + y^2}$ is the radial distance (m) from the centre of the PSF and D is the aperture diameter (m).

The detector PSF is used to simulate the blurring of the image within the detector element area [24, p. 89]. The detectors are assumed to have a 100% fill factor. The PSF of the detector is calculated using [2, pp. 462 and 465], [24, p. 89],

$$PSF_{\text{detector}}(x, y) = \frac{\text{rect}(x/p_x) \text{rect}(y/p_y)}{p_x p_y}, \quad (6.3.3a)$$

where

$$\text{rect}(u) = \begin{cases} 0, & \text{if } u > 0.5 \\ 0.5, & \text{if } u = 0.5 \\ 1, & \text{if } u < 0.5, \end{cases} \quad (6.3.3b)$$

where p_x and p_y are the detector dimensions (m) and $\text{rect}(u)$ is the rectangular top-hat function.

The electronic PSF is used to model the bandwidth shaping of the noise reduction electronic filters. The effect of the filters is projected onto the optical image plane [2, p. 90]. Low-pass Butterworth-type filters are commonly used in whiskbroom scanners to smooth the data in the across-track direction [2, p. 90]. This type of filter, however, is not common in pushbroom CCD sensors. In the case of pushbroom sensors, the electronic PSF is used to model sensor array properties, such as the CCD charge transfer efficiency [2, p. 90]. For this study the electronics PSF was not included in the PSF blurring kernel.

The WorldView-2 satellite telescope has an aperture diameter of 1.1 m and a focal length of 13.3 m [44], [46], [48], [51]. The WorldView-2 optics has an f -number of $f'/D = 12.09$. The optics obscuration ratio is assumed to be 0.3 [52]. Using the WorldView-2 detector and optics specifications and (6.3.2) and (6.3.3), the sensor PSF is calculated. Figure 6.3 illustrates the shape of the normalised optics, detector and total sensor PSF curves for the panchromatic band.

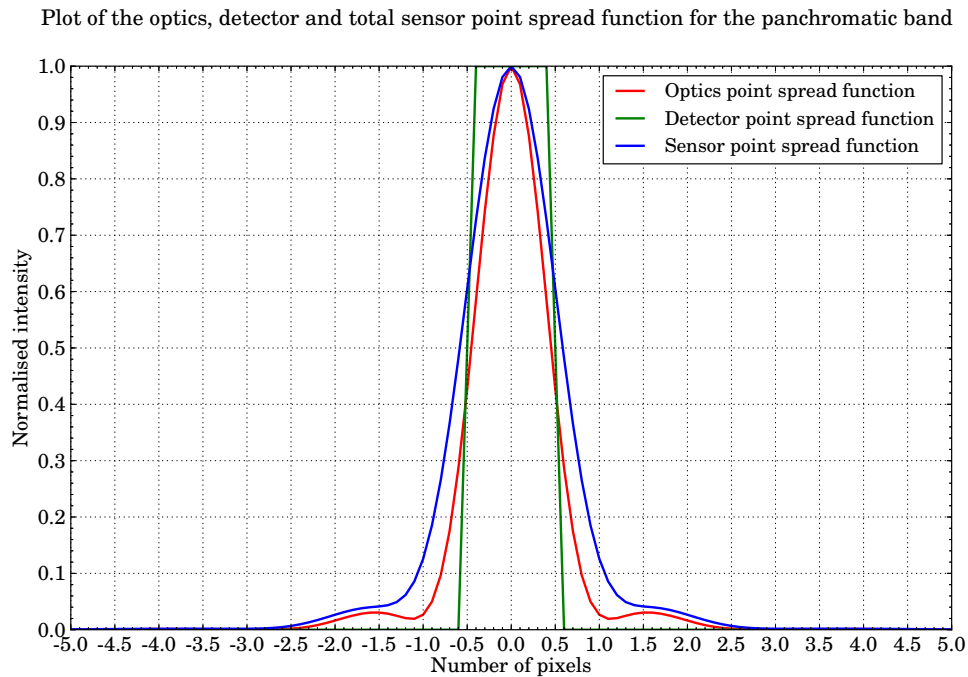


Figure 6.3: Shape of the point spread function for the kernel used to convolve with the output image to simulate the spatial blurring due to the sensor optics and detector.

6.3.4 Charged coupled device focal plane layout

A CCD array configuration can be a single continuous line of detectors, a staggered linear array or a rectangular 2D array as shown in Figure 2.10 in section 2.2.1.4. The generic sensor model designed in this study was able to model any one of these detector array configurations, provided no detectors in the array overlap. The CCD array configuration is specified in an XML data file.

The sensor model CCD array size for each imaging spectral band is set by specifying the number of detector element rows and columns. The sensor model divides the CCD array into even and odd DSA groups. The number of detectors in each DSA is calculated by dividing the number of CCD detectors in an array by the number of required DSAs. This must be an exact integer division with no remainder allowed. The even or odd numbered DSA groups can then be staggered along or across the focal plane. The whole CCD array for an imaging spectral band can also be shifted along or across the focal plane. The detectors are staggered and shifted relative to the Cartesian origin of the focal plane. Figure 6.4 illustrates the sensor model CCD array configuration for a row of detectors shifted and staggered along the focal plane.

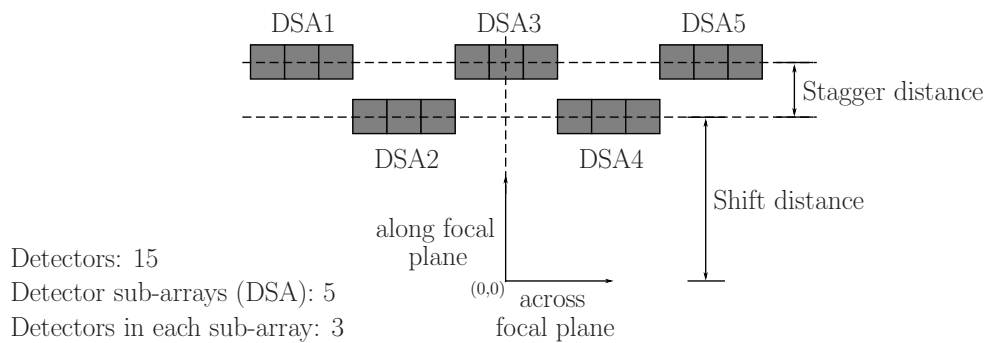


Figure 6.4: Charged coupled device array configuration illustrating the along-focal plane stagger and shift distances.

The focal plane coordinate of each detector is used to calculate the point in object space to be imaged onto the detector. The adjusted positions of all the detector elements should not exceed the bounds of the high resolution image. The detector focal plane position for an unstaggered and unshifted array is calculated and the adjusting distance values are then added to this position. Listing 6.1 provides the pseudo code used to calculate the focal plane distance by which to adjust each detector.

WorldView-2 uses a silicon CCD array with a detector pitch of $8\mu\text{m}$ for the panchromatic array and $32\mu\text{m}$ for the multispectral array [21], [44], [46]. Hence, with a focal length of 13.3 m an at-nadir ground sample distance (GSD) of 0.46 m is achieved when operating in the panchromatic mode and 1.85 m when operating in the multispectral


```

//-----
//Calculate the along and across focal plane distance by which to adjust each
//detector.
//-----
//--- Check that the number of detectors is exactly divisible by the number of
//    DSA groups. Return the number of detectors in each DSA group.
CheckExactDivision(totalNumberOfDetectors, numSubArrayGroups,
                   numberDetectorsInDetectorSubArrays, exactlyDivisible);

//--- Calculate the distance by which to adjust each detector
for(row=0; row<number of detector rows in the sensor array; row++)
{
    for(col=0; col<number of detector columns in the sensor array; col++)
    {
        //Determine the DSA group (currentSubArrayGroup) for detector(row,col).
        getDSAGroup(detectorPos, numberOfDetectorInSubArray, currentSubArrayGroup)

        //Stagger if the detector is in the stagger DSA group (even or odd)
        if(currentSubArrayGroup == staggerSubArrayGroup)
            useStagger = userSetStagger;
        else
            useStagger = 0.0;

        //Shift and stagger the detector by the user defined distances
        adjustBy = useStagger + userSetShift;
    } //End loop over columns
} //End loop over rows

```

Listing 6.1: C++ pseudo code used to calculate the along and across focal plane stagger and shift distances for each detector in the detector array.

mode [21], [44], [46], [47], [51]. The panchromatic detector array consists of 35 420 detectors and each of the multispectral detector arrays consists of 8881 detectors. The WorldView-2 sensor, therefore, has an at-nadir swath width of 16.4 km.

The WorldView-2 sensor multispectral detector arrays are grouped into two multispectral groups which are arranged on either side of the panchromatic detector array, in the focal plane. The first multispectral group consists of the NIR-1, red, green and blue detector arrays. The second multispectral group consists of the red edge, yellow, coastal blue and NIR-2 detector arrays [21]. The WorldView-2 satellite CCD array focal plane configuration is illustrated in figure 6.5.

To model the WorldView-2 CCD array configuration, the square detector pitch is set to be equal to the WorldView-2 detector pitch of 8 μm for the panchromatic band and 32 μm for the multispectral bands.

For the purposes of this study, it was assumed that no detectors between the different DSAs overlapped. Based on this assumption and in order to model an at-nadir swath width of 16.4 km, the panchromatic band CCD array was set to consist of 35 400 detectors and each of the multispectral bands to have 8850 detectors. These values allow the

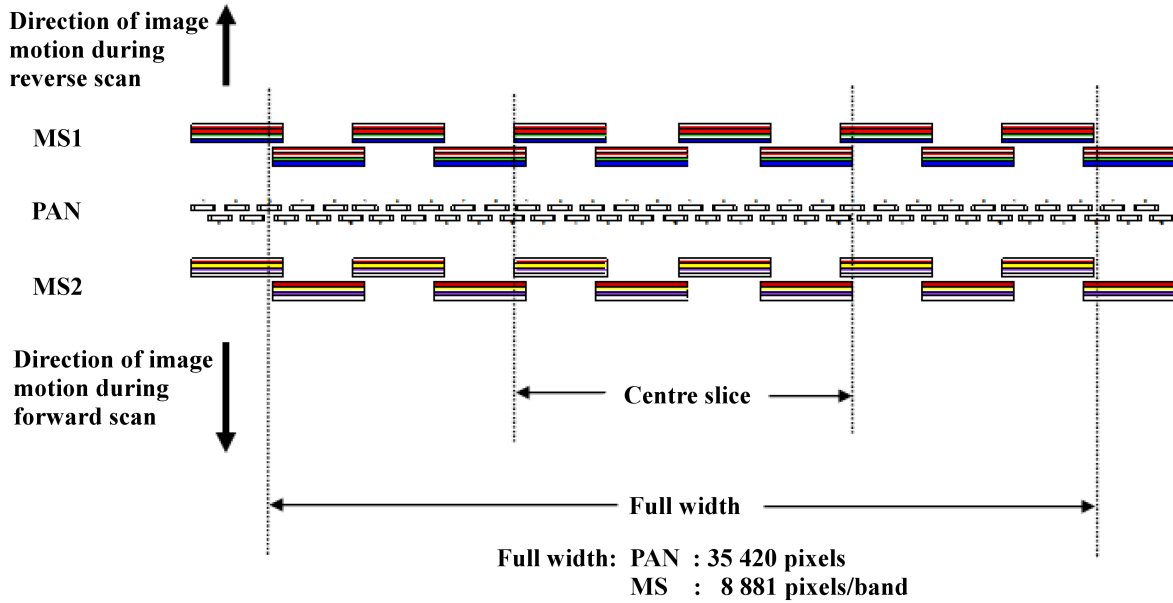


Figure 6.5: WorldView-2 satellite sensor focal plane array layout. MS1 is the first multispectral detector array group in the order of NIR-1, red, green and blue. PAN is the panchromatic detector array and MS2 is the second multispectral array group in the order of red edge, yellow, coastal blue and NIR-2 (image sourced from [21]).

detectors in the panchromatic CCD array to be divided into 50 DSAs and the detectors in the multispectral band CCD arrays to be divided into 10 DSAs, which matches the number of DSAs in the WorldView-2 CCD arrays.

It is assumed that the detectors in the CCD arrays are staggered and shifted relative to one another by multiples of the detector size. This assumption is imposed by the high resolution image implementation, discussed in section 10.4. The stagger distance between the DSAs is therefore set to 0.768 mm (24 multispectral pixels) for the multispectral band CCD arrays and 0.512 mm (64 panchromatic pixels) for the panchromatic band CCD array. The shift distance is set to 32 μm between the multispectral band CCD arrays. The shift distance between the panchromatic band CCD array and the adjacent multispectral band CCD arrays, is set to 96 μm . These values were chosen so that the shift and stagger effect would be large enough to be evaluated easily and still ensure that all detectors would remain within the bounds of the high resolution scene image.

The maximum size of the high resolution image limits the stagger and shift distances of the CCD array. This high resolution image size is limited by the available computer memory and execution time. This constraint is significant if the full Worldview-2 image is computed, but for many practical investigations a smaller-sized FOV can be defined with no loss of utility, as discussed in section 10.4.

For the WorldView-2 satellite sensor model, no shift or stagger takes place in the

across-track direction. If an across-track scanner is being modelled, then the across-track stagger and shift values can be set.

6.3.5 Detector dwell time

The time required for the sensor to form the signal for a single pixel is known as the detector integration time or dwell time. The dwell time is calculated such that in one time interval the sensor line of sight will have moved by one ground-projected instantaneous field of view (GIFOV), or GSD, along the scan direction [15, p. 14]. This allows the scene to be sampled with minimal motion blur, overlapping lines, or gaps in the output image.

If a detector is not sufficiently sensitive, the required signal-to-noise ratio (SNR) is not achieved within the maximum dwell time allowed. TDI, as described in section 2.2.1.4 and Figure 2.11, can then be used to improve the SNR [15, p. 15]. TDI offers a SNR improvement equal to the square root of the number of integration elements [1, p. 252]. For this study TDI was not included in the sensor model and is set as future work.

The sensor sample parameters are related to each other by [15, p. 14]

$$D_g = v \Delta t_s, \quad (6.3.4a)$$

and

$$\text{GIFOV} = \frac{p H}{f'}; \quad (6.3.4b)$$

setting $D_g = \text{GIFOV}$ results in

$$\Delta t_s = \frac{p H}{v f'}, \quad (6.3.4c)$$

where v is the satellite platform velocity (m/s), Δt_s is the integration time (s), p is the pixel size (m) in the focal plane, GIFOV is the pixel size (m) on the ground and D_g is the ground sample spacing (m) in the flight direction.

The WorldView-2 satellite platform velocity is calculated to be 7471.89 m/s, using (5.3.1), for the orbiting altitude of 770 km [21], [44], [46], [47], [51].

Using (6.3.4), a dwell time of 62 μ s is calculated for the panchromatic band and 248 μ s for the multispectral bands.

6.3.6 Detector noise

The image quality in all sensors is degraded by detector and electronics noise [9, p. 203]. The sensor model accounts for these noise sources by adding Gaussian distribution random number sequences to the pixel signals [24, p. 194]. The random number generators available in OSSIM, listed in Appendix A.5, are used to implement the satellite sensor noise.

The sensor noise is expressed using the noise equivalent irradiance (NEE) (W/m^2) [9, p. 258]. NEE is defined as the optical irradiance required to create an electronic signal

such that the SNR is one [9, p. 140]. In this study, however, the detector noise for the WorldView-2 sensor is not known, but estimates of the noise equivalent reflectance ($NE\Delta\rho$) are available. $NE\Delta\rho$ (unitless) is the change in reflectance required to create an electronic signal such that the SNR is one [9, p. 259]. The sensor NEE value is therefore calculated using the available $NE\Delta\rho$ specification, the noise equivalent exitance (NEM) and a scaling factor ($dM/d\rho$) with the units W/m^2 . Equations (6.3.5a) to (6.3.5c) provide the $NE\Delta\rho$ to NEE conversion for a Lambertian diffuse reflecting surface [9, p. 246];

$$NEE = \frac{NEM \omega_d}{\pi}, \quad (6.3.5a)$$

and

$$\frac{dM}{d\rho} = \frac{NEM}{NE\Delta\rho},$$

$$\therefore NEM = \frac{dM}{d\rho} NE\Delta\rho, \quad (6.3.5b)$$

by substitution of (6.3.5b) into (6.3.5a),

$$NEE = \frac{\omega_d}{\pi} \frac{dM}{d\rho} NE\Delta\rho \quad (6.3.5c)$$

where ω_d is the detector solid angle FOV, calculated as $\omega_d \approx (p_x p_y)/(f'^2)$, since the pixel IFOV is small [9, pp. 241-242]. p_x and p_y are the detector dimensions (m). The calculated NEE is used as the input standard deviation to the noise random number generator, with a mean of zero.

The WorldView-2 noise characteristics were not available but were assumed to be the same as those of the IKONOS and WorldView-1 satellites [53], [54]. The $NE\Delta\rho$ values are calculated using the sensor SNR. The SNR values for IKONOS and WorldView-1 are given in the satellite product guides ([53] and [54]) and are specified at half the quantisation range. A full quantisation range of 1228 greylevels (410 gl to 1638 gl) is assumed for this study, as described in section 6.3.8 [24, p. 87]. Using the SNR, the photon noise can be calculated as $noise = DN_{50\%}/SNR$ (gl), where $DN_{50\%} = 614$ gl, which is 50% of the 1228 gl quantisation range. $NE\Delta\rho$ can then be calculated as $NE\Delta\rho = noise/DN_{100\%}$, where $DN_{100\%} = 1228$ gl, which is the full quantisation range. Table 6.1 provides a list of the assumed SNR and calculated $NE\Delta\rho$ values, for each of the WorldView-2 imaging spectral bands,

The scaling factor $dM/d\rho$ value is estimated using (6.3.5c) and the photon noise and $NE\Delta\rho$ values given in Table 6.1. The estimated $dM/d\rho$ value is then adjusted slightly in the simulation environment, if required, to ensure that the desired SNR is achieved for the specified $NE\Delta\rho$ value.

The sensor noise values are specified in the model XML configuration file.

The pixel NEE value from (6.3.5c) is converted into a greylevel pixel noise value (P_{noise}), which is used in (6.3.6) in section 6.3.8. To convert the pixel NEE value into the P_{noise} value, the pixel NEE value is multiplied by the detector fill factor and optics

Table 6.1: WorldView-2 calculated noise equivalent reflectance values, for each of the imaging spectral bands. The 50% quantisation range is set to 614 gl.

Spectral band	Signal to noise ratio at 50% quantisation range (unitless)	Photon noise (gl)	Noise equivalent reflectance (unitless)
Panchromatic	237	2.5907	0.0021
Coastal blue	313	1.9617	0.0016
Blue	313	1.9617	0.0016
Green	317	1.9369	0.0016
Yellow	317	1.9369	0.0016
Red	324	1.8951	0.0015
Red edge	324	1.8951	0.0015
NIR-1	311	1.9743	0.0016
NIR-2	311	1.9743	0.0016

entrance pupil area (m^2) to get the pixel noise flux (W). The pixel noise flux is then multiplied by the detector responsivity value (gl/W), as derived in section 6.3.8, to get the value of P_{noise} (gl).

6.3.7 Detector non-uniformity

Detector element response (gain and offset) variations result in structured non-uniformities in the raw image pixel values [1, p. 595], [21]. In a pushbroom system, the detector non-uniformity in the CCD arrays results in dark and light streaks or bands along the image [21]. Variations between different detector elements result in image streaks one pixel wide. Variations between different readout registers result in image streaks as wide as the number of detector elements read out by the register. This study considers only detector element non-uniformity. The non-uniformity due to the readout register has been set as future work.

The WorldView-2 non-uniformity values were not available, therefore a non-uniformity gain (P_{NUgain}) of 2% and offset (P_{NUoffset}) of 1 gl are assumed; these values are used in (6.3.6) in section 6.3.8. The values are based on a typical photoresponse non-uniformity peak-to-peak value of 5% and a dark signal non-uniformity of 2 mV as specified in datasheets [55], [56], [57]. The root mean square value of the photoresponse non-uniformity peak-to-peak value is calculated as $5\%/2\sqrt{2} = 1.77\%$ which is rounded up to give the estimated non-uniformity gain of 2%. The offset greylevel is calculated as $(2048 \text{ gl}/7 \text{ V}) \times 2 \text{ mV} = 0.59 \text{ gl}$, which is rounded up to give the estimated offset greylevel. This calculation assumes a saturation condition of a 2048 gl output for a maximum voltage of 7 V [56].

The sensor's non-uniformity values were specified in the model's XML configuration file, in terms of the statistical spread in gain non-uniformity and offset. These values are used as inputs to a random number generator.

6.3.8 Analogue-to-digital conversion

The detector output electronic signal is processed and undergoes analogue-to-digital conversion before transmission to a ground station [1, p. 167]. The electronic signal is processed by adding an offset and multiplying by a gain factor. This is done so that the signal optimally spans the range required for analogue-to-digital conversion [1, p. 167], [2, pp. 106-107]. The signal amplification should not allow the signal to saturate [2, p. 106]. The amplified signal is sampled and quantised into DN's [2, p. 107]. The generic sensor model provides the functionality to calculate the greylevel value of a pixel, based on the definition of the gain, offset and ADC resolution.

The WorldView-2 satellite sensor employs 11-bit ADC quantisation [21], [44], [46], [47]. The simulated WorldView-2 sensor model theoretically has 2048 greylevels available. Most sensors do not use the full quantisation range. Safety margins are set at the lower and upper limits to prevent the output from saturating on occasionally high or low signal events, noise events or when doing non-uniformity correction. For this study an upper and lower greylevel safety margin of 20% of the full 2048 gl range is arbitrarily chosen which results in the full quantisation range being set as 410 gl to 1638 gl. The simulation model allows the user to set the upper and lower ADC limits in the configuration file. To achieve this range the detector signal is adjusted by gain and offset processing.

The pixel greylevel is calculated using,

$$P_{gl} = \left[(G_{\Phi_p} \Phi_p + C_{\Phi_p}) R_{gl} + P_{noise} \right] P_{NUgain} + O_{gl} + P_{NUoffset}, \quad (6.3.6)$$

where P_{gl} is the pixel greylevel value (gl), P_{noise} is the pixel noise (gl), O_{gl} is the sensor lower DN offset (gl) and P_{NUgain} and $P_{NUoffset}$ are the detector non-uniformity gain and offset (gl), respectively. Φ_p is the pixel flux (W), C_{Φ_p} (W) and G_{Φ_p} (unitless) are the flux offset and gain values respectively and R_{gl} is the detector responsivity in terms of greylevels (gl/W).

The pixel flux in (6.3.6) is calculated from the at-sensor radiance ($W/(m^2 \cdot sr)$) from (4.3.3), excluding the in-FOV adjacency effect, in section 4.3. The radiance is converted into flux (W) by multiplying the radiance by the solid angle of the detector IFOV (sr), the detector fill factor (unitless) and the optics collection area (m^2). The pixel flux is then adjusted to account for any offset and gain changes caused by the atmosphere. The flux offset is equivalent to the atmospheric path radiance and the flux gain is the change observed in the gradient of the ideal flux against reflectance graph when an atmosphere is added, as calculated by,

$$G_{\Phi_p} = \frac{\Phi_{p,\rho=1} - \Phi_{p,\rho=0}}{\Phi_{p,\rho=1} - C_{\Phi_p}}, \quad (6.3.7)$$

where $\Phi_{p,\rho=1}$ is the pixel flux for a 100% reflective target. $\Phi_{p,\rho=0}$ is the pixel flux for a 0% reflective target with no atmosphere present, which results in $\Phi_{p,\rho=0} = 0$. The G_{Φ_p} and C_{Φ_p} values are calculated for the atmospheric conditions discussed in section 5.3.1.

The adjusted flux is then multiplied by the detector responsivity. The user must supply the detector responsivity in terms of greylevels per watt of flux in the optical entrance aperture. The detector responsivity is calculated using

$$R_{gl} = R_{\lambda} G_{gl}, \quad (6.3.8)$$

where R_{gl} is the detector responsivity in terms of greylevels (gl/W), R_{λ} is the detector responsivity (A/W) and G_{gl} is the greylevel amplification gain factor (gl/A). The G_{gl} value for each spectral band is selected such that an object with 100% reflectivity produces a greylevel of 1638 at the sensor output, when no noise or detector non-uniformity is present. The detector responsivity for each imaging spectral band was calculated using python code, as illustrated by the pseudo code snippet in Listing 6.2.

```

#--- Model the silicon detector response (A/W) with
#   lambdaC=1.15um, cuton= 3.5, cutoff=4.3, scaling=8.
for wav in bandWavelength:
    siDetRes = scaling*((wav/lambdaC)**cuton - (wav/lambdaC)**cutoff)

#--- Calculate the system response (unitless) using the band spectral
#   response and the normalised detector response.
systemRes = Slambda*siDetRes/siDetResPeak

#--- Calculate the pixel flux using the at-sensor solar and path radiance.
#   W = [W/(m^2.sr)+W/(m^2.sr)]*sr^2*unitless*m^2*unitless
pixFlux = (SolarRadianceAtSen + PathRadianceAtSen)*(detectorIFOV**2)
         *fillFactor*collectArea*systemRes

#--- Adjust the pixel flux for the atmosphere.
#   W = (W-W)*unitless
pixFluxAdjust = (pixFlux - fluxOffset)*fluxGain

#--- Calculate the detector responsivity using the expected greylevel output.
pixDet = pixFluxAdjust*siDetResPeak # A = W*A/W
gain = expectedGl/pixDet # gl/A and expectedGl=1638gl
responsivity = gain*siDetResPeak # gl/W = (gl/A)*(A/W)

```

Listing 6.2: Python pseudo code snippet used to calculate the detector responsivity.

The greylevel offset, detector noise and non-uniformity are then taken into account in (6.3.6). The value of O_{gl} is set such that an object with 0% reflectivity produces a greylevel of 410 at the sensor output.

Table 6.2 lists the flux offset and gain values and the calculated detector responsivity values.

Table 6.2: List of flux offset and gain factors and detector responsivity values, for each of the WorldView-2 imaging spectral bands.

Spectral band	Pixel flux gain (unitless)	Pixel flux offset (W)	Detector responsivity (gl/W)
Panchromatic	1.090 470	$1.162\ 330 \times 10^{-12}$	$8.767\ 70 \times 10^{13}$
Coastal blue	1.282 850	$8.625\ 231 \times 10^{-12}$	$3.139\ 22 \times 10^{13}$
Blue	1.200 420	$9.531\ 550 \times 10^{-12}$	$2.151\ 02 \times 10^{13}$
Green	1.128 480	$8.160\ 830 \times 10^{-12}$	$1.713\ 21 \times 10^{13}$
Yellow	1.109 967	$3.854\ 960 \times 10^{-12}$	$2.860\ 70 \times 10^{13}$
Red	1.082 340	$4.894\ 075 \times 10^{-12}$	$1.908\ 78 \times 10^{13}$
Red edge	1.069 790	$2.271\ 070 \times 10^{-12}$	$3.528\ 35 \times 10^{13}$
NIR-1	1.047 920	$3.894\ 512 \times 10^{-12}$	$1.441\ 83 \times 10^{13}$
NIR-2	1.043 630	$2.102\ 843 \times 10^{-12}$	$2.441\ 27 \times 10^{13}$

The quantisation process results in uncertainty in the signal level, since a range of values may produce the same DN. This uncertainty is known as quantisation noise [1, p. 593], [24, p. 86]. The inclusion of quantisation noise to this model is set as future work.

6.4 Conclusion

A generic satellite sensor model was designed that allows a user to simulate different sensor configurations for passive remote sensing. The model considers the definition of each imaging spectral band in the sensor separately. This allows the different sensor spectral bands to have unique values for the same setting, for example, the detector size for the panchromatic band CCD is different from that of a multispectral band CCD.

The sensor model includes a telescope optics model. This model simulates the spectral reflectivity of the four mirrors used in the optics.

Spatial blurring in the output image, which is due to the sensor optics and detectors, is included in the sensor model by means of a PSF blurring kernel. This PSF kernel is convolved with the idealised high resolution image to obtain the sensor output image.

The sensor model includes a configurable CCD array model that can model a number of different CCD array layouts for both along-track and across-track scanners. These scanners are discussed in Chapter 7. The CCD array is divided into DSA groups which can be shifted and staggered in the focal plane. The detector noise and non-uniformity is included in the CCD array model by means of random number generators. The CCD array model also allows the user to set the dwell time of the imaging spectral bands.

The sensor model includes an ADC. The ADC is used to produce a DN representation of the sensor output image.

Chapter 7

The Satellite Swath Scanning

7.1 Introduction

The satellite scanning motion converts the spatial at-sensor radiance into a continuous time-varying optical signal, which is collected by the optics and sampled by the detectors [2, p. 76]. A satellite sensor forms an image of a scene by using either along-track or across-track scanning methods [3, p. 326]. These scanning methods are illustrated in Figure 2.9 in section 2.2.1.4.

Along-track scanners, such as pushbroom scanners, make use of the satellite's forward motion to scan the terrain. As the satellite moves along the track, the detector array samples successive image lines, perpendicular to the flight direction [3, p. 332]. The WorldView-2 satellite makes use of a pushbroom scanner [44].

Across-track scanners use rotating or oscillating mirrors to scan the detector array footprint sideways across the satellite ground track [1, p. 194], [3, p. 326]. Paddlebroom scanners make use of rotating mirrors to scan the detector footprint in one direction across the track [2, p. 19]. Whiskbroom scanners make use of oscillating mirrors to scan the detector footprint in both directions across the track [1, p. 203], [2, p. 19]. As the scanning mirror rotates or oscillates, the satellite platform advances along the track, this allows the scanner to sample successive lines along the ground track [1, p. 194], [3, p. 326]. This type of scanning method will, however, result in gaps and overlapping regions in the ground coverage [1, p. 194]. A scan line corrector, such as that used by the Landsat satellites, can be used to correct this error by shifting the scan projection on the ground so that successive scan lines fall alongside each other [1, p. 204]. The scan line corrector consists of two rotating parallel mirrors which advance and retard the forward and reverse scan lines. The scan line is shifted to start slightly ahead of the satellite across-track location and end behind the starting point. This prevents gaps or overlapping regions from occurring in the ground coverage [1, p. 204].

7.2 The generic satellite scanning model

The generic satellite sensor model must be able to simulate both along-track and across-track scanning configurations. A Landsat-type scan line corrector, however, is not within the scope of this study.

Along-track scanning is simulated by adjusting the satellite attitude and position as a function of time, synchronised with the detector sampling interval. The satellite attitude is adjusted so that the sensor's line-of-sight points down towards the location of the terrain to be sampled during that time interval. The satellite position is adjusted to move the satellite along the track once during every detector sample interval. This allows a single across-track image line to be sampled each time the sensor moves one GSD along the track.

The across-track scanning motion is simulated by first adjusting the satellite's body attitude to pitch the satellite in such a way that the position of the sensor centreline is aligned in the along-track direction. Secondly, the sensor's line-of-sight is adjusted in the across-track direction (sideways), relative to the satellite body. The sensor's line-of-sight is initially adjusted to point towards one end of the swath scan. The sensor's line-of-sight direction is then incrementally adjusted across the track at successive sampling intervals and by doing this it simulates the across-track scanning motion. The satellite position and attitude is adjusted to move one GSD along the track each time the full width of the swath has been scanned.

7.3 Implementing the satellite along-track scanning configuration model

OSPA implements a function to calculate the incremental changes in attitude (roll, pitch and yaw) and the Cartesian position for a given satellite trajectory. These incremental changes are used in the generic OSSIM delta-dynamics libraries, to calculate the observer object attitude and position at each successive detector sampling time interval [7, p. 140]. The delta-dynamics library adjusts the attitude and position of an observer successively by a user specified amount.

For along-track scanning, the satellite's initial attitude is set to point at nadir and for this study, it remains unchanged during the simulation run. The satellite attitude is adjusted using a yaw-pitch-roll rotation sequence. The yaw angle is used to adjust the satellite relative to the scan direction, so that the sensor detector array is aligned perpendicular to the ground track. The satellite is pitched to point downwards towards the earth's surface. The roll angle is adjusted to align the sensor to the flight direction.

The delta dynamics functionality is then used to set the increment value by which

the satellite's Cartesian position coordinate along the flight direction is adjusted. By adjustment of the satellite position by a fixed increment, the satellite is made to move at a constant rate in the along-track direction. The sensor is then able to image one swath line each time the satellite moves forward.

The satellite attitude, position and forward motion settings must be set up correctly in the model configuration files in order for the along-track scanner functionality to operate in the desired manner. For an along-track scanner, the satellite is set to yaw by 0° to align the sensor across the track, pitch by 90° to look downwards to the target and to roll by 180° for a descending orbit. These attitude angles will not change for the duration of the simulation run. The satellite along-track position will increment by one GSD while the satellite across-track position remains at the centre. The satellite altitude will also remain constant.

7.4 Implementing the satellite across-track scanning configuration model

The across-track scanning satellite model required the addition of a gimbal to steer the pointing mirror across the track. The gimbal functionality allows the user to adjust the sensor line-of-sight relative to the object body [7, p. 142]. OSPA includes an across-track scan mirror angle algorithm to calculate the scan mirror attitude angles to correspond with each detector sampling time interval. The calculated attitude angles are applied to the gimballed mirror to adjust the sensor's line-of-sight. The along-track movement of the satellite is modelled using the OSSIM delta-dynamics libraries, as discussed in section 7.3.

The across-track scanning functionality provides the ability to simulate either an oscillating scan mirror or a rotating scan mirror. The starting attitude for successive scan cycles is determined according to the type of scanning mirror being simulated. The scan mirror attitude for each across-track scan line is adjusted relative to the starting scan mirror attitude.

For a rotating mirror (sawtooth scan), the initial attitude of the mirror for each scan is always the same. This ensures that each swath line is scanned in the same direction across the track.

For an oscillating mirror (triangular scan), the initial attitude of the mirror for the next scan is always the end attitude of the previous scan. This ensures that successive swath lines are scanned in alternating directions across the track.

The algorithm to adjust the gimbal angle, and hence the mirror attitude, is summarised by the pseudo code in listing 7.1

In listing 7.1 `deltaCounterEnd` defines the number of times the image is sampled in one swath scan. To calculate the value of `deltaCounterEnd`, the user must supply the swath

```

//-----
//This algorithm adjusts the scan mirror attitude for each time interval.
//-----
//--- Determine if the attitude must be reset or adjusted
if(deltaCounter == deltaCounterEnd) //End of the scan line
{
    //Reset the counter
    deltaCounter = 0;

    if(rotatingScanMirror) // Scan in the same direction
    {
        //The starting attitude is set to the initial values.
        startingAttitude(roll, yaw, pitch) = initialAttitude(roll, yaw, pitch);
    }

    if(oscillatingScanMirror) // Scan in the opposite direction
    {
        //Set the starting point to the end attitude value of the previous line.
        startingAttitude(roll, yaw, pitch) = previousAttitude(roll, yaw, pitch);

        //Set the scan direction to be in the opposite direction
        deltaSign *= -1;
    }

} //End if(deltaCounter == deltaCounterEnd)
else //The end of the scan line has not been reached
    deltaCounter += 1;

//--- Calculate the new attitude
attitude(roll, yaw, pitch) = startingAttitude(roll, yaw, pitch)
    + (deltaCounter*deltaSign*deltaAttitude(roll, yaw, pitch));

```

Listing 7.1: C++ pseudo code used to calculate the scan mirror attitude for the current simulation time.

width FOV (θ_{sw} in rad) and the across-track sweeping angle increment ($\Delta\theta_s$ in rad). The number of times the image is sampled in one scan cycle (ΔC_{end}) is then calculated, using

$$\Delta C_{end} = \frac{\theta_{sw}}{\Delta\theta_s}. \quad (7.4.1)$$

Figures 11.9, 11.10, 11.12 and 11.13, in sections 11.7 and 11.8, illustrate the satellite's trajectory and the gimballed scan mirror's attitude when the satellite is set to use an across-track scanning configuration model. For an across-track scanner the satellite is set to yaw by 0° to align the sensor along the track, pitch by 90° to look downwards to the target and to roll by 180° for a descending orbit. These attitude angles will not change for the duration of the simulation run.

7.5 Conclusion

The OSSIM object behaviour and gimbal libraries and the OSPA across-track mirror angle algorithm are used to simulate the motion of along-track and across-track scanning.

The OSSIM delta-dynamics libraries were used to calculate the satellite along-track trajectory for each simulation time interval. The satellite attitude is set to point downwards towards the target. The satellite attitude remains constant for the full simulation run. No additional functionality is required to simulate an along-track scanner. However, for across-track scanners an across-track mirror sweep functionality is required.

The OSPA across-track scanning mirror algorithm is able to simulate the motion of rotating scan mirrors, as well as oscillating scan mirrors. The algorithm calculates the gimbal angles and, consequently, the scan mirror across-track attitude values.

The along-track and across-track attitude calculations are time-synchronous with the detector array sampling intervals.

The forward motion of the satellite along the track is defined by the satellite's orbit trajectory. Chapter 8 describes the OSPA orbit trajectory calculation.

Chapter 8

The Satellite Trajectory

8.1 Introduction

The satellite orbit and attitude determine the view the satellite will have of the earth [25]. When choosing an orbit the capability and objectives of the satellite sensor, the satellite motion, earth rotation and sun-earth-satellite orientation angles should be considered [1, p. 49], [3, p. 393], [8, p. 36]. The orbits considered in this study are for low orbiting earth observation satellites.

It is desirable to place an earth observation satellite in an orbit that allows the sensor to maintain an acceptable spatial resolution and that will periodically image all the points of interest on the surface of the earth when in nadir view [1, p. 49]. The temporal resolution (time interval) with which the satellite sensor images the same point on earth, is known as the satellite revisit time [2, p. 76], [4, p. 2.42], [8, p. 44].

Sun-synchronous orbits are a special case of LEO orbits, as discussed in section 2.2.1.4, which are retrograde, near-polar orbits which allow earth observation satellites to achieve the desired spatial and temporal resolution [1, p. 49], [16, p. 163], [25], [29]. These orbits allow a satellite to always cross the equator at the same local solar time [25]. In so doing, a consistent sun-earth angle is maintained, which results in images of an area, taken in the same season, to have the same illumination conditions. This allows for the images to be compared easily, without having to take into consideration extreme changes in shadows and lighting [25]. The near-polar inclination angle of the orbit will allow the sensor to image almost all locations on the surface of the earth [16, p. 163]. LEO orbits are generally near-circular orbits, which allow the sensor to achieve a constant image scale [1, p. 49], [2, p. 110], [3, p. 394], [29].

The satellite ground track is used to describe the satellite's orbit. The ground track traces the nadir path of the satellite over the surface of the earth [16, p. 165], [29]. The ground track is used to determine when certain locations on the earth will pass into the sensor FOV and when an observer will be able to see the satellite [16, p. 153], [29].

Atmospheric drag and gravity alter the orbit of an earth observation satellite [25], [29]. As a result, the satellite orbit is adjusted regularly to maintain the desired sun-synchronous orbit [25].

8.2 The satellite orbit calculation model

A simple satellite orbit trajectory model was designed to calculate the satellite's position and attitude, for a user-defined satellite orbit. The model provides the functionality to calculate a full orbit trajectory around the earth or a trajectory path between two points on the earth's surface. Only circular orbits were considered in this study and the orbit velocity is assumed to remain constant throughout the simulation. The effects of the sun-earth orbit, atmospheric drag, and gravity, on the satellite orbit are not included in the model. The satellite orbit trajectory model provides a framework that can be expanded to include elliptical orbits and the effect of external forces.

The Keplerian orbital elements are used to describe fully the satellite's orbit in space and the position of the satellite within the orbit [16, pp. 155 and 165], [58]. There are six Keplerian orbital elements to consider, namely, the eccentricity (e), the semi-major axis (a), the inclination angle (i), the right ascension of the ascending node (RAAN) angle (Ω), the argument of perigee angle (ω) and the true anomaly angle (v) [16, p. 160], [29], [58]. The Keplerian orbital elements are illustrated in Figure 8.1.

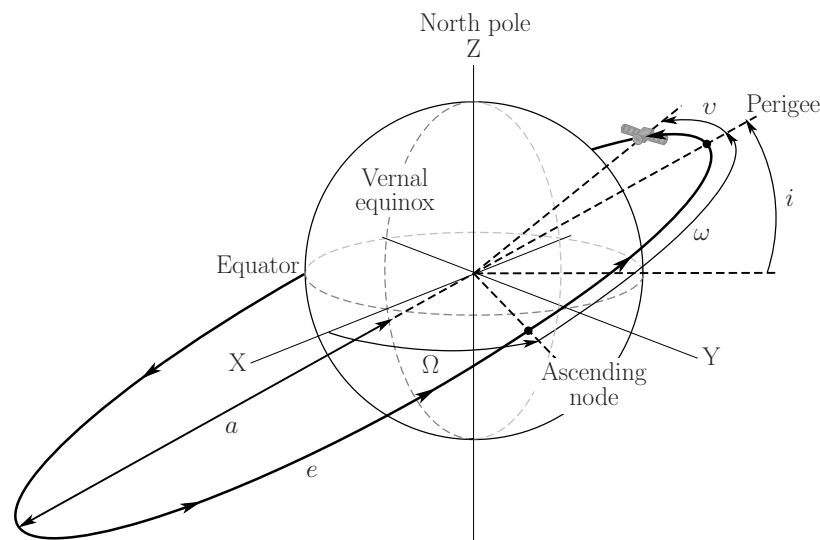


Figure 8.1: Illustration of the Keplerian orbital elements on an elliptical orbit: eccentricity (e), semi-major axis (a), inclination angle (i), right ascension of the ascending node angle (Ω), perigee angle (ω) and true anomaly angle (v) (adapted from [16, p. 160] and [58])

As illustrated in Figure 8.1, the orbit's eccentricity describes the amount by which the satellite orbit's shape deviates from a perfect circle [16, p. 156], [25]. The semi-major

axis defines the size of the orbit. The semi-major axis is half the length of the longest diameter of the orbit [16, p. 155]. The orbit inclination angle describes the orientation of the orbit with respect to the equator [3, p. 394], [16, p. 156], [29], [58]. The RAAN angle is used to describe the orientation of the orbit with respect to the vernal equinox. The RAAN angle is measured eastwards along the equator from the vernal equinox to the ascending node [16, p. 158]. The orbit's ascending node is the point where the satellite crosses the equator from the southern hemisphere into the northern hemisphere [29]. The argument of perigee angle defines the location of the orbit's perigee within the orbit plane [16, p. 158], [58]. The argument of perigee angle is measured along the orbit path from the ascending node to the orbit's perigee. This angle is measured in the direction of the satellite's motion [16, p. 158]. The true anomaly angle is used to describe the position of the satellite in an orbit. This angle is measured in the direction of the satellite's motion from the orbit's perigee point [16, p. 159].

The satellite orbit trajectory model calculates the satellite trajectory based on the properties of a circular satellite orbit. This study uses the WorldView-2 satellite orbit. WorldView-2 has a sun-synchronous, near-circular orbit. WorldView-2 orbits at an altitude of 770 km, with an inclination angle of 97.8° . WorldView-2 has an orbital period of 100.2 minutes and a 10:30 a.m. descending node crossing time [21, p. 1], [44], [47], [59].

To set the orbit to a near-polar retrograde orbit, the orbit inclination is set to between $90^\circ < i \leq 180^\circ$ [16, p. 157]. When calculating the circular orbit, the model assumes an eccentricity value of zero and the semi-major axis value is equivalent to the orbit radius [3, p. 394], [16, p. 156]. The orbit radius is defined as the distance from the centre of the earth to the satellite's position above the earth. The model does not include an argument of perigee angle or a true anomaly angle, since these angles are not present in a circular orbit [16, pp. 159-160].

8.3 Implementing the satellite orbit calculation model

OSPA requires a trajectory file that describes the satellite's position and attitude as a function of time, in order to move the satellite along the desired orbital path. An OSPA orbit model is designed to calculate the necessary orbit trajectory data to write to the trajectory file. The orbit trajectory calculation requires defining the orbit in the earth geographic coordinate system, then converting to the Earth-Centred-Earth-Fixed (ECEF) coordinate system and, finally, converting to a Local-Level coordinate system. The conversion to the Local-Level coordinate system is necessary because OSSIM uses a Local-Level coordinate system, as mentioned in Appendix A.6. The satellite's position is described in the Local-Level North-East-Down (NED) coordinate system. The satellite's

attitude is given as Euler angles in a yaw-pitch-roll sequence.

The orbit's trajectory is calculated in OSPA for the set of user-specified orbital parameter values, provided in an XML configuration file. The satellite orbit trajectory algorithm requires a description of the orbit, the planet radius around which the satellite is orbiting and the reference latitude and longitude coordinates. The orbit trajectory to be calculated is defined as being either a full circular orbit around an earth-sphere target, or a direct path between two sets of latitude and longitude coordinates on the earth-sphere target. The earth-sphere target is a spherical target used to represent the earth. This target is described in Appendix C.2.6.

The full circular orbit trajectory path is described by the orbit altitude, inclination angle and RAAN angle. The trajectory path between two points is described by the orbit altitude and the start and end latitude and longitude coordinates, on a perfectly spherical earth. The trajectory path is calculated relative to a reference coordinate expressed in earth geographic coordinates. The trajectory values are calculated at regular time intervals, as specified by the user. The position coordinates along the trajectory path are evenly spaced and maintain a constant altitude above the surface of the earth-sphere target. By default, the full orbit is calculated for a single cycle around the earth-target. The full orbit can, however, be calculated to make the satellite orbit a number of times around the target. This is useful if the user wishes to simulate the revisit time of the satellite, when a *rotating* earth-sphere target is used.

8.3.1 Calculating the satellite position coordinates

The orbit trajectory is calculated in a number of steps. All calculations are performed in a right-handed coordinate system. The superscript]^G denotes the earth geographic coordinate system (latitude (φ) and longitude (λ)). The superscript]^E denotes the ECEF coordinate system (x, y, z). The superscript]^L denotes the Local-Level coordinate system (x, y, z). The orbit is calculated relative to a user defined latitude and longitude reference point.

A WorldView-2 satellite type orbit that passes over the Pretoria area is used to demonstrate the calculation steps. The orbit trajectory is set to be descending with an inclination angle of 97.8° and a RAAN angle of 212° . The reference coordinate is set to be at the top of the earth-sphere target ($90^\circ 0' 0''\text{N}$, $0^\circ 0' 0''\text{E}$) at an altitude of zero. The reference coordinate is rewritten as $(\varphi_{\text{ref}}, \lambda_{\text{ref}}) = (\pi/2 \text{ rad}, 0 \text{ rad})$. Two consecutive coordinates ($0^\circ 0' 0''\text{N}$, $25^\circ 44' 54.59''\text{E}$) and ($0^\circ 0' 0''\text{N}$, $25^\circ 45' 7.96''\text{E}$) along the orbit trajectory path, are chosen to be used to demonstrate the coordinate conversion process and the satellite attitude calculation steps. These coordinates are rewritten as $(\varphi_1, \lambda_1) = (0 \text{ rad}, 0.449 396 05 \text{ rad})$ and $(\varphi_2, \lambda_2) = (0 \text{ rad}, 0.449 460 85 \text{ rad})$. The satellite orbit trajectory position coordinates are calculated as follows:

1. An initial satellite trajectory path is calculated as latitude and longitude coordinates in the geographic coordinate system $]^G$. The coordinates are calculated as either a direct path between two points or as a circular path around the target.

1.1. To calculate the trajectory path between two coordinates, the latitude and longitude coordinates' increment size along the path must be known. The increment size is calculated as being the distance between the two coordinates divided by the number of coordinates required to be generated between these points. The coordinate increment value is added a number of times to the starting coordinate until the end coordinate is reached. This algorithm is summarised by the pseudo code in listing 8.1.

```
//-----
//Calculate the coordinates between the start and stop coordinates. Assume
//that the path between the two coordinates is a direct line.
//-----
//--- Get the latitude and longitude coordinate increment size.
incrementSizeLat = (startCoordlat - stopCoordlat) / numberOfIntervals;
incrementSizeLong = (startCoordlong - stopCoordlong) / numberOfIntervals;

//--- For each interval get the latitude and longitude coordinate
for(int i=0; i<=numberOfIntervals; i++)
{
    latitudeCoord.push_back(startCoordlat - (i*incrementSizeLat));
    longitudeCoord.push_back(startCoordlong - (i*incrementSizeLong));
} //End for(int i=1; i<=numberOfIntervals; i++)
```

Listing 8.1: C++ pseudo code used to calculate the latitude and longitude trajectory path coordinates between two points.

1.2. The full circular orbit trajectory is calculated in two steps.

1.2.1. The coordinates for an *equatorial orbit* are first specified by setting the latitude value to zero for all points in the trajectory. The longitude value is specified as a vector of values from 180° to -180° for an ascending orbit and -180° to 180° for a descending orbit. $C_{\text{equatorial}}^G(\varphi, \lambda)$ is used to represent the geographic latitude (φ) and longitude (λ) equatorial coordinates. From this equatorial orbit, two consecutive coordinates $(\varphi_1, \lambda_1) = (0 \text{ rad}, 0.449\,396\,05 \text{ rad})$ and $(\varphi_2, \lambda_2) = (0 \text{ rad}, 0.449\,460\,85 \text{ rad})$ are selected to further illustrate the calculation process.

1.2.2. The orbit RAAN and inclination angles are used to rotate and tilt the equatorial orbit coordinates to produce the required satellite ground track trajectory coordinates $(C_{\text{satellite}}^G(\varphi, \lambda))$. This coordinate transformation

is done by first converting the $C_{\text{equatorial}}^G(\varphi, \lambda)$ coordinates to (x, y, z) equatorial coordinates using,

$$x^G = r_{\text{EARTH}} \cos(\varphi) \cos(\lambda), \quad (8.3.1a)$$

$$y^G = r_{\text{EARTH}} \cos(\varphi) \sin(\lambda), \quad (8.3.1b)$$

$$z^G = r_{\text{EARTH}}(1 - e^2) \sin(\varphi), \quad (8.3.1c)$$

where r_{EARTH} has the value 6378.1 km and the eccentricity is set to $e = 0$ for a circular orbit. Using (8.3.1) the coordinate $(\varphi_1, \lambda_1) = (0 \text{ rad}, 0.449 396 05 \text{ rad})$ is converted into $(x_{\text{equatorial}}^G, y_{\text{equatorial}}^G, z_{\text{equatorial}}^G) = (5744816.126, 2770784.597, 0.0)$.

The (x, y, z) equatorial coordinates are then rotated and tilted to calculate the desired satellite ground track coordinates $(x_{\text{satellite}}^G, y_{\text{satellite}}^G, z_{\text{satellite}}^G)$ using,

$$\begin{bmatrix} x_{\text{satellite}}^G \\ y_{\text{satellite}}^G \\ z_{\text{satellite}}^G \end{bmatrix} = \begin{bmatrix} x_{\text{equatorial}}^G \\ y_{\text{equatorial}}^G \\ z_{\text{equatorial}}^G \end{bmatrix} \mathbf{R}_{zx}^G, \quad (8.3.2)$$

where \mathbf{R}_{zx}^G is the rotation matrix defined as [60, p. 30],

$$\begin{aligned} \mathbf{R}_{zx}^G &= \mathbf{R}_z^G \mathbf{R}_x^G \\ &= \begin{bmatrix} \cos(\Omega_{\text{rot}}) & \sin(\Omega_{\text{rot}}) & 0 \\ -\sin(\Omega_{\text{rot}}) & \cos(\Omega_{\text{rot}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(i_{\text{tilt}}) & \sin(i_{\text{tilt}}) \\ 0 & -\sin(i_{\text{tilt}}) & \cos(i_{\text{tilt}}) \end{bmatrix} \\ &= \begin{bmatrix} \cos(\Omega_{\text{rot}}) & \sin(\Omega_{\text{rot}}) \cos(i_{\text{tilt}}) & \sin(\Omega_{\text{rot}}) \sin(i_{\text{tilt}}) \\ -\sin(\Omega_{\text{rot}}) & \cos(\Omega_{\text{rot}}) \cos(i_{\text{tilt}}) & \cos(\Omega_{\text{rot}}) \sin(i_{\text{tilt}}) \\ 0 & -\sin(i_{\text{tilt}}) & \cos(i_{\text{tilt}}) \end{bmatrix}, \quad (8.3.3) \end{aligned}$$

where Ω_{rot} and i_{tilt} are the rotation and tilt angles, respectively.

\mathbf{R}_z^G in (8.3.3) is the z-axis rotation matrix, used to rotate the equatorial $]^G$ coordinates, so that the ascending node is at the longitude location corresponding to the RAAN angle. For an ascending orbit, the trajectory values are rotated clockwise by the angle $\Omega_{\text{rot}} = -\Omega$, around a downward pointing positive z-axis. For a descending orbit, the trajectory values are rotated anti-clockwise by the angle $\Omega_{\text{rot}} = (\pi - \Omega)$, around a downward pointing positive z-axis.

\mathbf{R}_x^G in (8.3.3) is the x-axis rotation matrix, used to tilt the equatorial $]^G$ coordinates so that the orbit has the correct inclination angle. For an ascending orbit, the trajectory values are rotated clockwise by the angle $i_{\text{tilt}} = \pi - i$. For a descending orbit, the trajectory values are rotated clockwise by the angle $i_{\text{tilt}} = i$.

Using the RAAN angle of 212° and the inclination angle of 97.8° for a descending orbit, the Ω_{rot} value is set to $-0.558\,507\,346\,4$ rad and the i_{tilt} is set to $1.706\,932$ rad. Using these values the $(x_{\text{equatorial}}^G, y_{\text{equatorial}}^G, z_{\text{equatorial}}^G) = (5744816.126, 2770784.597, 0.0)$ coordinate is rotated and tilted to produce the coordinate $(x_{\text{satellite}}^G, y_{\text{satellite}}^G, z_{\text{satellite}}^G) = (5071145.062, 2725399.991, -2745148.861)$. The satellite $]^G$ coordinates are then converted to the ground track latitude and longitude coordinates $C_{\text{satellite}}^G(\varphi, \lambda)$ using,

$$r^G = \sqrt{(x^G)^2 + (y^G)^2 + (z^G)^2}, \quad (8.3.4a)$$

$$\varphi = \arcsin\left(\frac{z^G}{r^G}\right), \quad (8.3.4b)$$

$$\lambda = \arctan\left(\frac{y^G}{x^G}\right). \quad (8.3.4c)$$

Using (8.3.4) an $r^G = 6378100$ value which is equivalent to r_{EARTH} and a value $C_{\text{satellite}}^G(\varphi, \lambda) = (-25.493^\circ, 28.255^\circ)$ is calculated from the $(x_{\text{satellite}}^G, y_{\text{satellite}}^G, z_{\text{satellite}}^G)$ coordinate. This coordinate illustrates that the defined orbit will pass over the Pretoria area.

2. The ground track trajectory latitude and longitude coordinates are then converted into the ECEF coordinate system $]^E$, at the satellite orbiting altitude, using [60, p. 31]:

$$x^E = (r_{\text{EARTH}} + H) \cos(\varphi) \cos(\lambda), \quad (8.3.5a)$$

$$y^E = (r_{\text{EARTH}} + H) \cos(\varphi) \sin(\lambda), \quad (8.3.5b)$$

$$z^E = [r_{\text{EARTH}}(1 - e^2) + H] \sin(\varphi), \quad (8.3.5c)$$

where the semi-major axis length is equal to $r_{\text{EARTH}} + H$ and the eccentricity is $e = 0$, for a spherical earth and circular orbit. For an orbiting altitude of $H = 770$ km the $C_{\text{satellite}}^G(\varphi, \lambda) = (-25.493^\circ, 28.255^\circ)$ coordinate is converted to the satellite $]^E$ coordinate $(x_{\text{satellite}}^E, y_{\text{satellite}}^E, z_{\text{satellite}}^E) = (5683363.257, 3054426.166, -3076555.947)$.

Similarly, using (8.3.5) the earth geographic latitude and longitude reference coordinate ($90^\circ 0' 0''\text{N}$, $0^\circ 0' 0''\text{E}$) on the surface of the earth ($H = 0$), is converted into a ECEF reference coordinate $(x_{\text{ref}}^E, y_{\text{ref}}^E, z_{\text{ref}}^E) = (0, 0, 6378100)$.

3. The ECEF satellite $]^E$ coordinates are converted to the Local-Level NED $]^L$ coordinates using [60, p. 32],

$$\begin{bmatrix} x_{\text{NED}}^L \\ y_{\text{NED}}^L \\ z_{\text{NED}}^L \end{bmatrix} = \begin{bmatrix} x_{\text{satellite}}^E - x_{\text{ref}}^E \\ y_{\text{satellite}}^E - y_{\text{ref}}^E \\ z_{\text{satellite}}^E - z_{\text{ref}}^E \end{bmatrix} \mathbf{R}_{yz}^E. \quad (8.3.6)$$

where \mathbf{R}_{yz}^E is the rotation matrix and x_{NED}^L , y_{NED}^L and z_{NED}^L are the Local-Level NED coordinates. The rotation matrix \mathbf{R}_{yz}^E is derived using [60, pp. 30 and 32],

$$\begin{aligned} \mathbf{R}_{yz}^E &= \mathbf{R}_y^E \mathbf{R}_z^E \\ &= \begin{bmatrix} \cos\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) & 0 & -\sin\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \\ 0 & 1 & 0 \\ \sin\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) & 0 & \cos\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \end{bmatrix} \begin{bmatrix} \cos(\lambda_{\text{ref}}) & \sin(\lambda_{\text{ref}}) & 0 \\ -\sin(\lambda_{\text{ref}}) & \cos(\lambda_{\text{ref}}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \cos(\lambda_{\text{ref}}) & \cos\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \sin(\lambda_{\text{ref}}) & -\sin\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \\ -\sin(\lambda_{\text{ref}}) & \cos(\lambda_{\text{ref}}) & 0 \\ -\sin\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \cos(\lambda_{\text{ref}}) & \sin\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \sin(\lambda_{\text{ref}}) & \cos\left(-\frac{\pi}{2} - \varphi_{\text{ref}}\right) \end{bmatrix}. \end{aligned} \quad (8.3.7)$$

where \mathbf{R}_y^E and \mathbf{R}_z^E are the y-axis and z-axis rotation matrices, respectively. The NED coordinates are calculated relative to the reference latitude and longitude coordinate $(\varphi_{\text{ref}}, \lambda_{\text{ref}}) = (\pi/2 \text{ rad}, 0 \text{ rad})$. Using these values the coordinate $(x_{\text{satellite}}^E, y_{\text{satellite}}^E, z_{\text{satellite}}^E) = (5683363.257, 3054426.166, -3076555.947)$ is transformed into the $]^L$ coordinate $(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-5683363.257, 3054426.166, 9454655.947)$.

4. In the OSPA environment the coordinates calculated in the previous step result in an orbit around a point located at an altitude equivalent to the radius of the earth. To correct for this, the r_{EARTH} value is subtracted from the orbit altitude position (z_{NED}^L). This leads to the final orbit coordinate value of $(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-5683363.257, 3054426.166, 3076555.947)$ for the coordinate $(\varphi_1, \lambda_1) = (0 \text{ rad}, 0.44939605 \text{ rad})$.

Similarly, using these transformation steps, the $(\varphi_2, \lambda_2) = (0 \text{ rad}, 0.44946085 \text{ rad})$ example $]^G$ coordinate is calculated to have a final orbit coordinate value of $(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-568221.493, 3054270.914, 3076971.93)$.

8.3.2 Calculating the satellite attitude angles

The satellite attitude angles are calculated using the NED trajectory coordinates from (8.3.6). The satellite sensor line of sight is aligned with the body's positive x-axis, which is the principal plane direction. The satellite body is yawed and pitched to look towards the centre of the earth-sphere target and then rolled so that the sensor detector array is oriented correctly, relative to the track direction.

The satellite orbit attitude is calculated as follows:

1. The satellite yaw (ψ_{yaw}) and pitch (θ_{pitch}) attitude angles are calculated using,

$$r_{\text{NED}}^L = \sqrt{(x_c^L - x_{\text{NED}}^L)^2 + (y_c^L - y_{\text{NED}}^L)^2 + (z_c^L - z_{\text{NED}}^L)^2}, \quad (8.3.8a)$$

$$\psi_{\text{yaw}} = \arctan \left(\frac{(y_c^L - y_{\text{NED}}^L)}{(x_c^L - x_{\text{NED}}^L)} \right), \quad (8.3.8b)$$

$$\theta_{\text{pitch}} = -\arcsin \left(\frac{(z_c^L - z_{\text{NED}}^L)}{r_{\text{NED}}^L} \right). \quad (8.3.8c)$$

where r_{NED}^L is the slant distance between the NED position coordinate and the]^L coordinate at the centre of the earth $(x_c^L, y_c^L, z_c^L) = (0, 0, 0)$.

Equations (8.3.8) yield a satellite yaw angle of -0.4931435643 rad, pitch angle of 0.444938 rad and $r_{\text{NED}}^L = 7148100$, which is equivalent to $r_{\text{EARTH}} + H$, for the $(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-5683363.257, 3054426.166, 3076555.947)$ coordinate.

2. The satellite roll attitude angle is then calculated using three steps.

- 2.1. Firstly, the cross product of two consecutive coordinates along the orbit track is calculated using,

$$\vec{C}^L = \vec{A}^L \times \vec{B}^L, \quad (8.3.9a)$$

where

$$C_x^L = A_y^L B_z^L - A_z^L B_y^L, \quad (8.3.9b)$$

$$C_y^L = A_z^L B_x^L - A_x^L B_z^L, \quad (8.3.9c)$$

$$C_z^L = A_x^L B_y^L - A_y^L B_x^L. \quad (8.3.9d)$$

\vec{A}^L is the vector from the origin of the NED axes to the current position of the satellite, \vec{B}^L is the vector from the origin of the NED axes to the next position of the satellite and \vec{C}^L is the cross product's resultant vector. Each of the vectors has an x, a y and a z component, as shown by the subscripts.

Using the two example]^L coordinates

$$(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-5683363.257, 3054426.166, 3076555.947) \text{ and}$$

$$(x_{\text{NED}}^L, y_{\text{NED}}^L, z_{\text{NED}}^L) = (-568221.493, 3054270.914, 3076971.93)$$

$$\vec{A}^L = (-5683363.257, 3054426.166, 3076555.947)$$

and $\vec{B}^L = (-568221.493, 3054270.914, 3076971.93)$, \vec{C}^L is calculated to be $(1748212362, 2800327375, 449345841.3)$.

- 2.2. Secondly, the inclination angle between the two position coordinates (i_{AB}) is calculated as,

$$i_{\text{AB}} = \arccos \left(\frac{C_z^L}{r_c^L} \right). \quad (8.3.10)$$

where r_c^L is calculated using,

$$r_c^L = \sqrt{(C_x^L)^2 + (C_y^L)^2 + (C_z^L)^2}. \quad (8.3.11)$$

Using $\vec{C}^L = (1748212362, 2800327375, 449345841.3)$, i_{AB} is calculated to have a value of 1.435 512 796 rad for $r_c^L = 3331664982$.

2.3. Thirdly, the roll angle is calculated, using the inclination angle perpendicular to the track (i_\perp). i_\perp is calculated for i_{AB} using

$$i_\perp = \frac{\pi}{2} - i_{AB}. \quad (8.3.12)$$

The attitude roll angle (ϕ_{roll}) is then set according to the i_\perp value. For an inclination angle between 90° and 180° (retrograde orbit) or between 0° and 90° (prograde orbit), the ϕ_{roll} angle is set as,

$$\phi_{roll} = \begin{cases} i_\perp, & \text{for ascending prograde orbit} \\ -i_\perp, & \text{for ascending retrograde orbit} \\ \pi - i_\perp, & \text{for descending prograde orbit, } A_z^L < B_z^L \\ -(\pi + i_\perp), & \text{for descending prograde orbit, } A_z^L > B_z^L \\ \pi + i_\perp, & \text{for descending retrograde orbit, } A_z^L < B_z^L \\ -(\pi - i_\perp), & \text{for descending retrograde orbit, } A_z^L > B_z^L. \end{cases} \quad (8.3.13)$$

For an inclination angle of 0° or 180° (equatorial orbits), the ϕ_{roll} is set as,

$$\phi_{roll} = \begin{cases} -\frac{\pi}{2}, & \text{if } C_z^L > 0 \\ \frac{\pi}{2}, & \text{otherwise.} \end{cases} \quad (8.3.14)$$

Lastly, for an inclination angle of 90° (polar orbits), the ϕ_{roll} is set as,

$$\phi_{roll} = \begin{cases} \pi, & \text{for ascending polar orbit, } A_z^L \leq B_z^L \\ -\pi, & \text{for descending polar orbit, } A_z^L \leq B_z^L \\ 0, & \text{otherwise.} \end{cases} \quad (8.3.15)$$

The resultant roll angle should be adjusted to ensure that it lies within the range $[-\pi:\pi]$ rad.

The orbit used as an example is a retrograde orbit with $A_z^L < B_z^L$ which results in a satellite attitude roll angle of $\phi_{roll} = \pi + i_\perp = 3.276\ 876\ 184$ rad, where $i_\perp = \frac{\pi}{2} - 1.435512796 = 0.135\ 283\ 530\ 3$ rad. ϕ_{roll} is then adjusted to lie in the range $[-\pi:\pi]$ rad, $\phi_{roll} = 3.276876184 - 2\pi = -3.006\ 309\ 123$ rad.

8.4 Conclusion

The satellite orbit and attitude determine the view that the satellite has of the earth [25]. The six Keplerian orbital elements are used to fully describe a satellite orbit in space and the position of the satellite within the orbit [16, pp. 155 and 165], [58]. These elements are used in the satellite orbit trajectory model to describe the typical orbit followed by an earth observation satellite.

The orbit trajectory model is able to calculate a full circular orbit or a trajectory path between two points. The model assumes a circular orbit around a perfectly spherical earth and that no external forces on the satellite are present. The satellite trajectory for a Local-Level NED world coordinate system and a yaw-pitch-roll Euler rotation sequence, is calculated.

The sensor model output will include geometric discontinuities, due to the satellite scan motion and focal plane array configuration. In order to better analyse the data, an image correction algorithm is used to correct for the discontinuities. The image processing model is discussed in Chapter 9.

Chapter 9

Image Pre-Processing for Geometric Discontinuities

9.1 Introduction

The raw satellite data requires processing before a usable image is created [5]. Numerous digital image processing operations can be applied to the image data. These image processing operations include processes to format and correct data, digital enhancement to improve the interpretability of the image and algorithms to analyse and extract information from the image [1, p. 169], [5], [8, p. 147]. The image processing operations can be categorised into one or more groups; namely, pre-processing, image enhancement, image transformation and image classification and analysis [3, pp. 483-485], [8, p. 147].

Pre-processing, also known as image restoration and rectification, is used to correct an image for sensor and platform specific radiometric and geometric distortions [3, p. 483], [8, p. 149]. This study covered image restoration to correct for geometric discontinuities in the image caused by a staggered detector array and an oscillating scan mirror. Further image processing operations were beyond the scope of this study.

9.2 Basic image pre-processing model

The OSPA sensor model calculates the raw satellite image as temporally sampled by the detector arrays. The raw image is streamed from the satellite to the ground station. The detector array stagger and scanning mirror results in a raw image stream with displaced pixels. Pre-processing operations must, therefore, be applied to this data to correct the pixel displacement in order to reconstruct the image correctly.

An image correction model is implemented in OSPA, which serves as a base to which more advanced image processing algorithms can be added.

The image correction model consists of only two basic image pre-processing algorithms. The first algorithm corrects the raw data for a staggered array layout and the second algorithm corrects the image for alternating scan line directions. These two OSPA image pre-processing algorithms are fixed operations that assume that all preconditions required to process the image correctly have been met. The algorithms correct by whole-number pixel shifts and do not perform any re-sampling or interpolation of the pixel values. The algorithms assume that the satellite views the scene at nadir for the duration of the simulation run. This assumption, along with a number of preconditions, makes it possible to correct the image without resampling the data [61]. Many of the preconditions are met by virtue of a well-behaved satellite motion. The preconditions are listed below:

1. The satellite attitude, flight direction, altitude and velocity must remain constant throughout the simulation.
2. The detector's temporal sampling must maintain a fixed sample rate.
3. The sensor's CCD array must consist of a single row of detectors for along-track scanning or a single column of detectors for across-track scanning.
4. The sensor's DSAs must be staggered by a integer number of detector sizes.
5. The across-track scan distance must remain constant throughout the simulation.
6. The start and end across-track scanning sensor-attitude angles must be equidistant from the centre of the swath width.
7. The swath width must be an integer multiple of the detector element size.

9.3 Implementing the staggered line correction algorithm

A staggered detector array layout results in the even and odd DSAs imaging two different along-track locations in the scene at the same time. As a result, the data for a single along-track location are sampled into two raw image lines. To correct for this distortion, the data from the same along-track location are extracted from the two distorted raw image lines and joined to form a corrected image line. The DSA stagger distance is used to determine which two image lines contain data of the same location. Interleaving pixels are added to scan lines where there is insufficient data to form a complete scan line. This occurs for scan lines taken at the beginning and end of the scanned area. The interleaving pixels are filled with a user defined value [61].

To correct for the along-track staggered layout, the number of rows in the image by which the pixels are shifted must be known. The ground sample spacing in the flight direction (D_g) is calculated using (6.3.4a) in section 6.3.5. The value of D_g (m) is then

converted into a distance on the sensor focal plane (d_g) by using

$$d_g = \frac{D_g f'}{H}, \quad (9.3.1)$$

where f' is the imaging system focal length (m) and H is the orbit altitude (m). The number of image pixels (N_{pixel}) by which the DSAs are staggered, is then calculated as

$$N_{\text{pixel}} = \frac{d_{\text{st}}}{d_g}, \quad (9.3.2)$$

where d_{st} is the distance (m) in the focal plane by which the DSAs are staggered. N_{pixel} is used to determine which two raw image lines to use when reconstructing a specific image line. DSAs that are staggered ahead of others are referred to as the leading DSAs. The leading DSA pixel data appears N_{pixel} raw image lines before the lagging DSA pixel data. The correction algorithm employs a loop to access each pixel in a raw image scan line. The pixels in the corrected scan line are filled with the lagging and leading DSA pixels that correspond to the scan line location that is being reconstructed from the two raw lines. The corrected line is saved to an image file and the next line is processed.

The first N_{pixel} image lines do not have corresponding image lines with which to correct the scan line. This is due to the leading DSAs never imaging the scene at the point at which the lagging DSAs first image the scene. The last N_{pixel} image lines also do not have corresponding image lines with which to correct the scan line. This is due to the lagging DSAs never imaging the scene at the point at which the leading DSAs last image the scene. In the case where data are not available, interleaving pixels are used. Figure 9.1 illustrates the stagger line correction process.

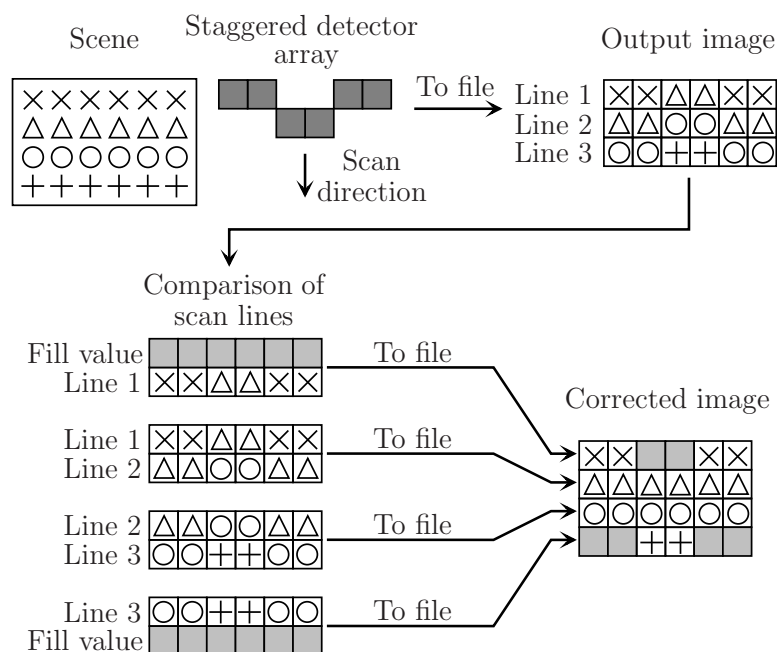


Figure 9.1: Staggered line image correction process.

```

//-----
//Correct the raw image for an along-track staggered detector array layout by
//constructing a corrected line using two raw image lines. Write the corrected
//line to a new file.
//-----
for(int line = 0; line < endOfLines; line++)
{
    //--- Fill the current scan line with image data
    frameCurrentToRead = line;
    if(line > (numberOfDetLinesInFile-1)) //There is no data in the file
        FillLineWith(lineToProcess1, fillWithUserValue);
    else
        FillLineWith(lineToProcess1, fillFromFile);

    //--- Fill the previous scan line with image data
    framePreviousToRead = frameCurrentToRead - staggerLineOffset;
    if(framePreviousToRead < 0) //There is no data in the file
        FillLineWith(lineToProcess0, fillWithUserValue);
    else
        FillLineWith(lineToProcess0, fillFromFile);

    //--- Construct the corrected scan line row
    for(int col = 0; col < numDetectorsInRow; col++)
    {
        if(usePixelFromPreviousLine)
            correctedLine(col) = lineToProcess0(col);
        else
            correctedLine(col) = lineToProcess1(col);
    } //End for(int col = 0; col < numDetectorsInRow; col++)

    //--- Write the new correct image line to the file
    correctedLine.WriteToFile(correctedImageFileName)
} //End for(int line = 0; line < endOfLines; line++)

```

Listing 9.1: C++ pseudo code used to correct the raw image for an along-track staggered detector array layout by constructing a corrected line using two raw image lines.

Listing 9.1 provides the pseudo code used to correct the image for an along-track staggered array layout.

Correcting staggered lines for an across-track scanner employs the same correction method as that used for the along-track scanner. In this case, however, the number of columns by which the image pixels are shifted is determined.

9.4 Implementing the alternating scan line correction algorithm

An oscillating scan mirror sweeps the detector sightline back and forth across the track. The back-and-forth scanning motion causes alternating across-track scans to have different scan directions. Each scan line along the swath is referred to as a frame. To correct the

alternating scan directions, the frames in the reversed scanned image lines are shifted until the image pixels align with those in the adjacent image lines. The correction algorithm assumes that the starting point of an image line aligns with the end point of the adjacent image lines. This assumption results in frames being shifted across the full width of the image line, and therefore no interleaving pixels being added to the edges of the across-track scan lines. This shifting is equivalent to reversing the scan direction of the frames. This correction does not correct for any additional pixels misaligned across the track due to the alternating scan line. A correction algorithm that shifts the pixels by an amount other than the full width of the image line, will be able to correct for any additional misaligned pixels. Such an alternating scan line correction algorithm has been set as future work and was beyond the scope of this study.

To correct for alternating across-track scans, each across-track frame is rewritten to a new image file, in the correct pixel order. The frames with the correct scan direction are written to the corrected image file in the same order as they appear in the raw image line. The frames with an incorrect scan direction are written to the file in the reverse order of how they appear in the image line. Figure 9.2 illustrates the alternating scan line correction process.

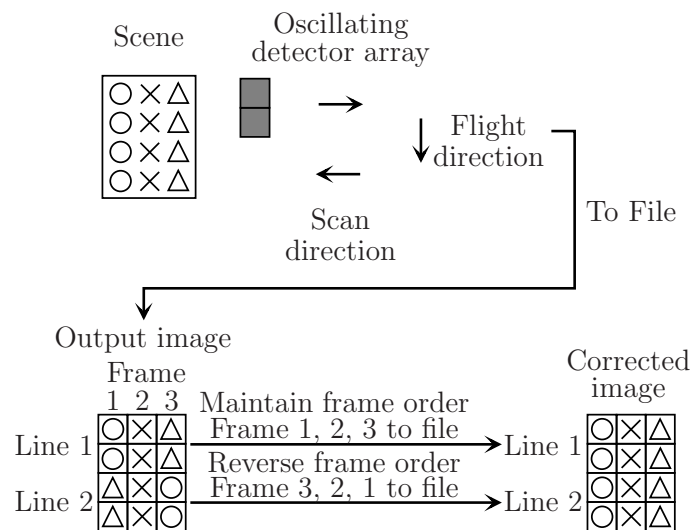


Figure 9.2: Alternating scan line image correction process.

Listing 9.2 provides the pseudo code used to correct the image for alternating scan line directions.

The detector array can be staggered while using an oscillating scan mirror to image the scene. The OSPA image processing algorithm is able to correct the image for both of these discontinuities. The algorithm will first correct for the oscillating motion and then for the stagger distortion across the track.

```

//-----
//Correct the raw image for alternating scan line directions by reversing the
//frame order of alternating lines. Write the corrected line to a new file.
//-----
for(int line=0; line < numberOfScanLinesInFile; line++)
{
    //--- Set the frame to be copied and set the flag if the line
    //      must be reversed
    if (line == lineToCorrect)
    {
        frame = linesToProcess.size()-1; //Start at the end of the line
        decFrame = true
    }
    else
    {
        frame = 0 //Start at the beginning of the line
        decFrame = false
    }

    //--- Write the frames to the file
    while(frameCounter < lineToProcess.size())
    {
        linesToProcess[frame].WriteToFile(correctedImageFileName)
        if(decFrame)
            frame--
        else
            frame++

        //Increase the frame counter
        frameCounter++;
    } //End while (frameCounter < lineToProcess.size())
} //End for (int line = 0; line < numberOfScanLinesInFile; line++)

```

Listing 9.2: C++ pseudo code used to correct the raw image for alternating scan line directions by reversing the frame order of alternating scan lines.

9.5 Conclusion

Raw satellite image data must be processed before a usable image can be generated from the data. There are a number of different processing operations that can be applied to the image data. Pre-processing operations are considered in this study. Pre-processing is used to correct an image for sensor and platform specific radiometric and geometric distortions. For this study, a basic image processing model was designed. This model is used to remove the geometric discontinuities in the image data that are due to a staggered detector array layout and an oscillating scan mirror, by shifting specific image pixel rows or columns by a set amount to realign the target geometry. A specific set of conditions must be met in order for the pre-processing algorithms to process the raw data correctly.

The OSPA user can expand this model to implement their own application-specific and advanced image processing algorithms.

Chapter 10

Implementation Issues and Limitations

10.1 Introduction

The main implementation issues when simulating a satellite remote sensing system are the hardware memory available and the computation time. The OSPA simulation environment must be able to simulate a satellite sensor accurately within a reasonable computational time while remaining within the hardware limitations. Issues arise when the size of data sets is restricted by the available hardware memory or results in a long computation time. These issues are addressed by limiting the size of the data sets used or by using an alternative method to simulate the desired effect.

The main limitation in the satellite remote sensing system model was the level of detail with which some features are modelled. For instance, the scope of the study did not include a TDI model or detailed models of the targets and satellite orbit. These limitations in the OSPA simulation environment can be addressed in future work by expanding on the models designed in this study.

10.2 Limitations of the real world scenes and targets

A detailed model of a real world target or scene must be geometrically accurate and requires extensive data on the radiometric properties of the target materials. Rigorous models require a specialist to help with scene characterisation. The design of such rigorous models is beyond the scope of this study, since the main focus was on the satellite sensor.

Basic target models were, however, designed and used in this study. This placed a limit on the scene and target details that could be imaged by the sensor. A description of

the targets used in this study are provided in Appendix C.

The design of highly detailed world scene models has been set as future work.

10.3 Implementation issues of the atmospheric aerosol scattering point spread function

The atmospheric scattering PSF, as discussed in section 5.3.4, is used to calculate a convolution kernel to model the atmospheric in-FOV adjacency effect that is due to aerosol scattering. The limitation when calculating the atmospheric scattering PSF is the dynamic resolution required when converting between the frequency and spatial domains. A large data set is required in order to obtain a sufficient resolution in both of the domains. The challenge is to have a sufficient resolution to be able to model the PSF centre spike as being a single pixel wide and also to model the PSF at distances further away from the centre spike, while keeping the data set size within the hardware constraints. These large data sets result in an out-of-memory failure, especially when a 2D data set is used. The large PSF kernel would also result in the convolution of the kernel with the image becoming a time consuming process.

The dynamic resolution and large data set required made it impractical to model the in-FOV adjacency effect using the PSF convolution kernel. To overcome this issue, the PSF kernel can be divided into a series of smaller kernels, which are applied many times. This method, however, becomes computationally time consuming when the kernel is applied to images produced from large detector arrays. One solution in this case would be to decrease the size of the CCD array, so that the desired effect can still be modelled within a reasonable computational time. A more suitable alternative would be to truncate the PSF and resample the subsampled kernel to match the sensor pixel intervals. Only the central portion of the 2D PSF kernel is used when truncating the kernel. The large size of the PSF, the comparatively smaller sensor array size and the gradual change in the magnitude of the PSF scattered curve over the sensor FOV, result in the subsampled kernel having an almost constant value over the kernel, except at the centre pixel where the PSF has a spike. Convolution with a constant value kernel, except at the centre spike, is equivalent to summing all the image pixels around the centre of the kernel and then multiplying the summed value by the constant kernel value and adding this value to the centre pixel. The calculated value is scaled to ensure that the condition is met that a no-gain kernel integral is unity. Another alternative solution would be to approximate the 2D PSF convolution kernel as an impulse with two one-dimensional convolution kernels, applied in orthogonal directions around the impulse. The one-dimensional convolution kernels would approximate the gradual change in the magnitude of the PSF scattered curve over the sensor FOV.

The in-FOV adjacency due to the aerosol scattering was modelled in this study by using a constant value. The PSF numerical model, as given in (5.3.13) in Chapter 5.3.4, was used to calculate a suitable constant value.

10.4 Limitations in the sensor charged coupled device focal plane layout

Satellite sensors use CCD arrays that consist of thousands of detectors that are shifted and staggered in the focal plane. In order for the detectors to image the scene in OSPA, the line of sight of the detector must remain within the bounds of the high resolution image. The CCD array size and the stagger and shift distances are therefore limited by the high resolution image size, as discussed in section 6.3.4 and Appendix A.4. The high resolution image size, in turn, is limited by the hardware memory available. Most effects of the CCD array configuration can, however, be demonstrated using a smaller CCD array and smaller stagger and shift distances. This would ensure that the available hardware memory was not exceeded and that the computation time for the simulation remained within acceptable limits.

The size of the detectors may differ between the different spectral bands of the sensor. For instance, the detectors used in the WorldView-2 panchromatic spectral band CCD array are smaller than those used in the multispectral spectral bands. The difference in the detector sizes must be accounted for in the high resolution image pixel size, which is a multiple of the detector's pixel size. The high resolution images used for the different sensor spectral bands are all set to have the same number and the same sized image pixels. The different detector sizes for the sensor must, therefore, be multiples of one another in order that the high resolution image pixels may be a multiple of all the different detector sizes for that sensor. This limitation in the high resolution image configuration also results in the limitation that the stagger and shift distances of the CCD arrays must be multiples of the detector size. These conditions ensure that the high resolution image is down convolved correctly to the sensor image, so that the image is rendered both geometrically and radiometrically correct when all the spectral bands are active in the simulation.

A solution to this limitation is to allow the high resolution image array dimensions for each spectral band to be independent of those of the other spectral bands. The high resolution image pixel size and number can then be set to best suit the CCD array configuration for that spectral band.

10.5 Limitations in the model due to time delay integration

The exclusion of a TDI model, as described in section 2.2.1.4, is a limitation in the sensor imaging method model. The inclusion of a TDI model has been set as future work.

A TDI model can be implemented by increasing the number of detector array rows, for along-track scanners, and columns, for across-track scanners, in a sensor spectral band detector array. The number of detector rows or columns added is determined by the number of TDI steps specified. As the sensor band scans across the scene, each detector along the scan direction will image the same swath line in the scene in turn. The data collected by each detector is then summed and averaged to get the signal for that scene location. This solution to increasing the number of detectors in the array will require more memory to model the spectral band CCD array and, as a result, will limit the implementable detector array size due to insufficient available hardware memory.

10.6 Limitations in the satellite orbit

A rigorous satellite orbit trajectory model was beyond the scope of this study. Only a basic satellite orbit trajectory was implemented, as described in Chapter 8. The trajectory path that can be simulated is limited to a path between two points, or a perfectly circular orbit around a spherical earth. More complex orbits are not available in the model. The basic orbit also does not include effects such as platform vibrations, orbit decay and the rotation of the earth around the sun. The orbit trajectory functionality serves as a framework on which a more detailed orbit model can be built. The expansion of the orbit capability has been set as future work.

10.7 Conclusion

The main limitations and issues in the OSPA simulation environment include the limited hardware memory available to store large data sets and perform calculations, the excessive computation time required and the less than ideal level of simulation detail in aspects such as the target and orbit. Issues regarding the hardware memory and time limitations depend on the computer used to perform the simulation runs and can be resolved by using smaller data sets, approximations where applicable or a more powerful computer. Limitations with regard to the level of simulation detail can be solved by expanding on the sensor and target models as future work.

Chapter 11

Validation of the Model

11.1 Introduction

As outlined in Chapter 3, the OSPA functionality is built on the OSSIM functionality and consists of the capabilities required to simulate a satellite remote sensing system. The functionalities must be validated to ensure that OSPA operates in the expected manner and produces the correct results. The OSPA functions designed and tested are listed:

1. The in-FOV adjacency effect due to atmospheric aerosol scattering.
2. A configurable CCD focal plane layout, used to simulate a staggered and shifted satellite CCD layout.
3. The $NE\Delta\rho$, used to calculate the NEE when simulating the sensor noise.
4. The ADC, used to convert the sensor output flux to a set of digital numbers.
5. The across-track scanner, which is used to simulate a rotating or oscillating mirror used in the satellite optics to scan along the swath width.
6. The satellite orbit trajectory model, which is used to calculate the satellite attitude and position for a trajectory path over the surface of a sphere.
7. The image pre-processing algorithm, which is used to correct the output image for geometric discontinuities due to the CCD layout and scanner.

OSSIM functionality that OSPA makes use of, such as the signature rendering equation, the OSSIM delta-dynamics libraries for forward motion, the MODTRAN atmosphere and detector non-uniformity, are not evaluated in this study. OSSIM is extensively tested and validated against Simulator for Imaging Systems (SIMIS), independent calculations and real world measurements [7], [62]. OSSIM is considered to be validated and trustworthy [7].

Test scenarios are created to evaluate each of the listed OSPA functionalities. The test scenarios describe the sensor settings, atmosphere, satellite motion and the target to be imaged. For this study the test scenarios make use of the available WorldView-2

satellite specifications. The simulated WorldView-2 CCD array configuration is set to be a single row of 35 400 detectors for the panchromatic band and 8850 detectors for each of the multispectral bands. Such large arrays result in long computational times, most of which are in excess of 20 minutes. The computational time to simulate a satellite scan of less than a second can take longer than 20 minutes. The model is, therefore, tested using fewer detectors in the arrays. The test scenarios validate the model within the limitations of the integrated OSPA simulation environment limitations. The test set provides full coverage of the OSPA functionality for the test scenarios.

A number of test targets have been designed to test specific functions of the satellite remote sensing system. These test targets as rendered in OSPA are shown in Figure 11.1. Note that for illustration purposes the images in Figure 11.1 have been scaled to have a maximum grey level of 255 and do not include any sensor or atmospheric effects. The radiometric and geometric properties of the targets are discussed in Appendix C.

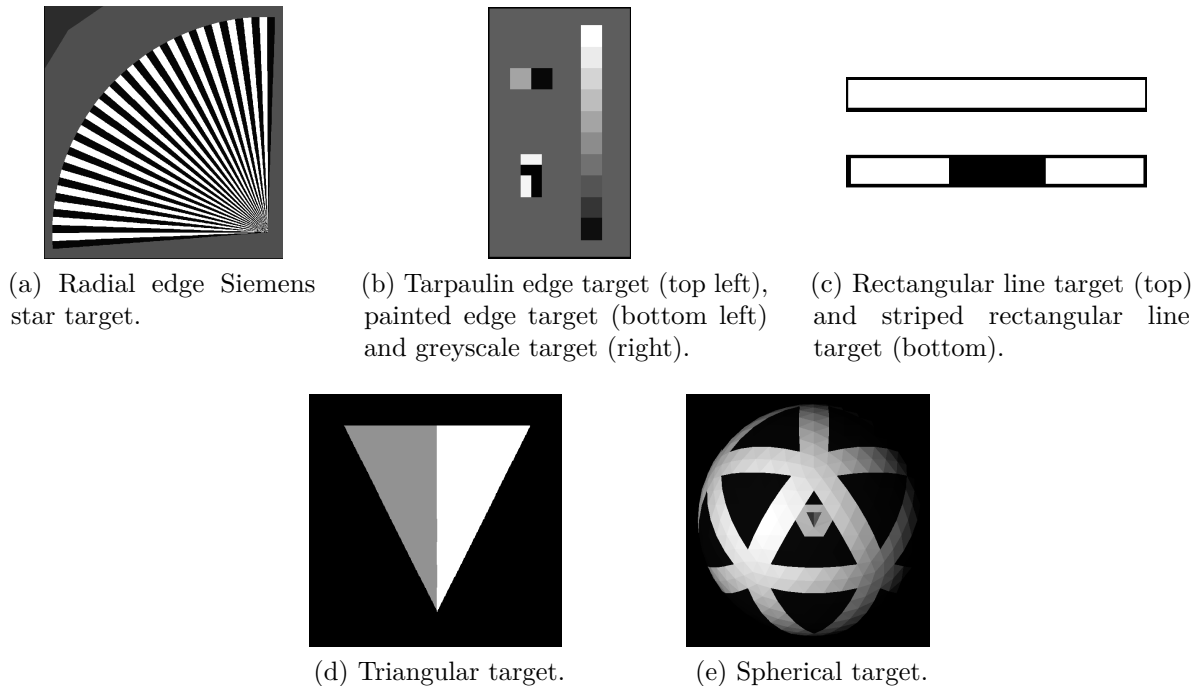


Figure 11.1: Test targets used to evaluate OSPA.

The OSPA test targets include the Siemens star in Figure 11.1a, to evaluate the MTF of the sensor output image and the greyscale bar and edge targets in Figure 11.1b, to evaluate the radiometric output of the system. The long narrow rectangular targets in Figure 11.1c and the triangular target in Figure 11.1d are used to evaluate the detector focal plane layout, SNR, the across-track scanning motion and the image correction algorithms. The spherical target in Figure 11.1e and the triangular target are used to evaluate the satellite orbit trajectory.

11.2 Atmospheric aerosol scattering for the in-field-of-view adjacency effect

11.2.1 Test description

This test validates the in-FOV adjacency effect term shown in (4.3.3) in section 4.3. This term was added to the OSSIM signature rendering equation (4.2.1) in order to account for the effect of flux from areas in and near the satellite sensor FOV on the target. The terms of the OSSIM signature rendering equation were not validated in this study, since the equation has been extensively tested in OSSIM, as discussed in [7] and [62].

As discussed in sections 4.3 and 5.3.4, the in-FOV adjacency effect term was implemented as a constant value with a peak value at the target pixel. The in-FOV adjacency effect was expected to reduce the image contrast and have a negligible effect to the system MTF.

To test the effect of the OSPA in-FOV adjacency effect term on the system MTF, the simulated satellite sensor was set to scan over a Siemens star target, Figure 11.1a, when the in-FOV adjacency effect was present. The CCD arrays were set to be a single row of 1000 detectors for the panchromatic band and 250 detectors for each of the multispectral bands. The CCD arrays were not staggered or shifted. No noise components such as detector non-uniformity and $NE\Delta p$ are present in this test. The sensor scanned the image using a pushbroom scanner.

The sensor output image of the star was used to calculate the system MTF curve by using the contrast value of the star segments at different radial distances from the star origin. Listing 11.1 shows how the MTF curve is calculated from a Siemens star image.

```
#--- Calculate the greylevel contrast at the radial distance
contrast = (glMax-glMin)/(glMax+glMin)

#--- Calculate the greylevel contrast for the full DN range
contrastFull = (upperDN-lowerDN)/(upperDN+lowerDN)

#--- Calculate the MTF as a percentage
MTF = (contrast/contrastFull)*100

#--- Calculate the cycles per radian
arcLength = radialDistance*GSD*siemensStarArcAngle
arcLengthToRad = 2*arctan2((arcLength/2.0),orbitAltitude)
cyclesPerRad = numberOfLinePairs/arcLengthToRad
```

Listing 11.1: Python code snippet used to calculate the modulation transfer function from a siemens star.

The aerosol scattering MTF, Figure 5.4 in section 5.3.4, has a much lower cut-off

frequency than other system MTF components and has an asymptotic value near atmospheric transmittance at that wavelength. As a result the MTF calculated from the Siemens star was expected to be determined by the instrumentation. For this test the system MTF was set to be equivalent to the detector MTF.

To test the in-FOV adjacency effect on the target contrast, the sensor was set to image a tarpaulin edge target, Figure 11.1b, with a reflectance of 55% and 3.5%. The simulated satellite was set to take a snapshot of the reflective targets. The CCD arrays were set to be a single row of 40 detectors for the panchromatic band and 10 detectors for each of the multispectral bands. The CCD arrays were not staggered or shifted. No noise components such as detector non-uniformity and $NE\Delta\rho$ were present in this test.

The calculation steps to determine the change in pixel irradiance due to the in-FOV adjacency effect are discussed, using the panchromatic imaging band as an example.

The OSSIM signature rendering equation was used to calculate the panchromatic band at-sensor radiance, without the aerosol scattering present, for the target reflectance. The irradiance values were then calculated by multiplying the at-sensor radiance values by the detector along and across track IFOV. The at-sensor pixel irradiance was calculated to be $1.661\,943 \times 10^{-12} \text{ W/m}^2$ for the 55% reflective block and $3.255\,887 \times 10^{-13} \text{ W/m}^2$ for the 3.5% reflective block. Using the plot in Figure 5.5 in section 5.3.4 the constant in-FOV adjacency value was calculated as being the average PSF intensity, excluding the centre peak, over the CCD array area. For the panchromatic band this value was calculated to be $2.52560489927 \times 10^{-6}$ (unitless) for two times the width of the WorldView-2 CCD array. This average PSF intensity is valid for both large and small CCD arrays since the PSF slopes gradually from the peak. The calculated constant value and a centre peak value of one were then scaled to ensure that the in-FOV adjacency effect over the PSF area was equal to one. The pixel area under consideration was set to 30×80 pixels which only covered the tarpaulin edge target area so that the adjacency effect was the result of only the two reflectance values of the target. The constant and centre peak value was scaled by a factor of $(2.52560489927 \times 10^{-6} \times 30 \times 79 + 1)^{-1} = 0.994$ to give a centre peak value of $1 \times 0.994 = 0.994$ and a constant PSF value of $2.52560489927 \times 10^{-6} \times 0.994 = 2.51058 \times 10^{-6}$.

The at-sensor irradiance value, including the in-FOV adjacency effect, was then calculated for each target reflectance value using 4.4.1,

$$\begin{aligned} I_{\text{inFovAdj},\rho=0.035} &= (2.51058 \times 10^{-6} \times [(1.661\,943 \times 10^{-12} + 3.255\,887 \times 10^{-13}) \times 30 \times 40]) \\ &\quad - (2.51058 \times 10^{-6} \times 3.255\,887 \times 10^{-13}) + (0.994 \times 3.255\,887 \times 10^{-13}) \\ &= 3.296\,22 \times 10^{-13} \text{ W/m}^2, \end{aligned}$$

and

$$\begin{aligned} I_{\text{inFovAdj}, \rho=0.55} &= (2.51058 \times 10^{-6} \times [(1.661943 \times 10^{-12} + 3.255887 \times 10^{-13}) \times 30 \times 40]) \\ &\quad - (2.51058 \times 10^{-6} \times 1.661943 \times 10^{-12}) + (0.994 \times 1.661943 \times 10^{-12}) \\ &= 1.657955 \times 10^{-12} \text{ W/m}^2. \end{aligned}$$

The OSSIM and OSPA irradiance values were analysed to determine the extent of the in-FOV adjacency effect on the image contrast. The change in contrast is most noticeable when a target of high reflectance is placed next to a target of low reflectance. The higher the contrast between the two targets the greater the in-FOV adjacency effect will be.

The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.2.2 Results

All simulated WorldView-2 spectral imaging bands imaged the Siemens star target. For illustration only the panchromatic image band results are plotted. Figure 11.2 shows the portion of the sensor output image (left) used to calculate the MTF and a section of the output image (right) that highlights the unresolved Siemens star pixels.

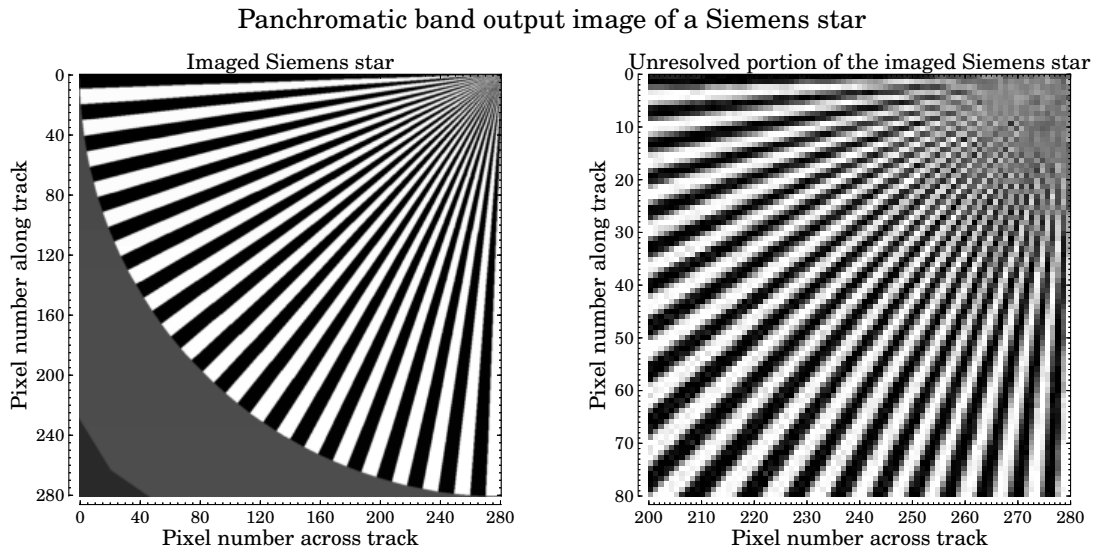


Figure 11.2: Panchromatic band output image of a Siemens star (left). The image on the right provides a closer look at the unresolved pixels of the image on the left.

The measured MTF was plotted using the contrast values calculated from the Siemens star image on the left in Figure 11.2. The calculated MTF is plotted in Figure 11.3.

The sensor focal length was 13.3 m and the panchromatic band detectors had a pitch of $8 \mu\text{m}$, which resulted in a MTF cut-off frequency of 1.66×10^6 cycles/rad. As seen in Figure 11.3, the system MTF matched the sinc shape and cut-off frequency of the square

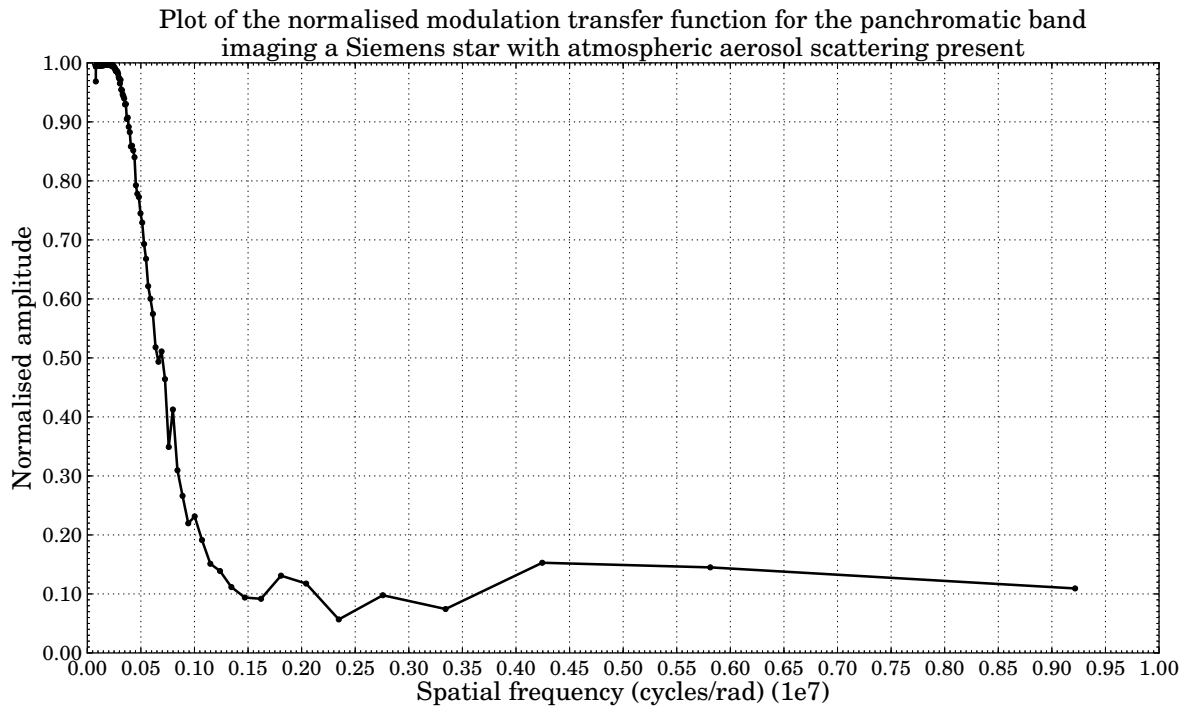


Figure 11.3: Plot of the modulation transfer function calculated from the sensor output image of a Siemens star. Atmospheric aerosol scattering is present.

detector's MTF curve. Similarly, the system MTF curves calculated for the multispectral band match the multispectral band detector MTF curves.

The constant and peak in-FOV adjacency values for the simulated WorldView-2 imaging bands are listed in Table 11.1.

Table 11.1: In-FOV adjacency effect constant and peak point spread function values scaled to the high resolution image area of 30×80 pixels for the panchromatic band and 30×20 pixels for the multispectral band.

Spectral band	Unscaled constant value ($\times 10^{-5}$) (unitless)	Unscaled peak value (unitless)	Scaled constant value ($\times 10^{-5}$) (unitless)	Scaled peak value (unitless)
Panchromatic	0.252 560 489 927	1	0.251 058	0.9940
Coastal blue	1.211 235 425 390	1	1.202 930	0.9931
Blue	1.123 704 374 290	1	1.116 553	0.9936
Green	1.035 390 870 400	1	1.029 316	0.9941
Yellow	1.009 063 344 260	1	1.003 293	0.9943
Red	1.015 374 201 640	1	1.009 531	0.9942
Red edge	0.730 010 546 707	1	0.726 986	0.9959
NIR-1	1.010 344 831 260	1	1.004 560	0.9943
NIR-2	1.000 702 552 020	1	0.995 027	0.9943

These values are used when calculating the expected pixel irradiance values in Tables 11.2 and 11.3. The tables list the OSSIM calculated pixel irradiance with no added in-FOV adjacency and the expected and OSPA calculated pixel irradiance when the in-FOV adjacency was added. The tables also list the change in irradiance due to the adjacency effect and the percentage error between the expected and the OSPA calculated irradiance value. The expected values are calculated as described in section 11.2.1.

Table 11.2 illustrates the in-FOV adjacency effect on a 3.5% reflective target adjacent to a 55% reflective target.

Table 11.2: In-FOV adjacency effect for a 3.5% reflective target adjacent to a 55% reflective target.

Spectral band	3.5% reflective target pixel irradiance ($\times 10^{-13}$)(W/m ²)			Change in irradiance (%)	Error (%)
	OSSIM (no in-FOV adjacency)	Expected (with in-FOV adjacency)	OSPA (with in-FOV adjacency)		
Panchromatic	3.255 887	3.296 22	3.296 480	1.231	-0.008
Coastal blue	1.134 670	1.141 65	1.136 351	0.148	0.464
Blue	1.313 711	1.323 72	1.316 149	0.185	0.572
Green	1.215 249	1.227 39	1.218 242	0.246	0.745
Yellow	0.599 471	0.606 56	0.601 218	0.291	0.881
Red	0.831 871	0.843 13	0.834 671	0.335	1.003
Red edge	0.412 961	0.417 49	0.414 080	0.270	0.817
NIR-1	0.869 813	0.886 37	0.873 934	0.472	1.403
NIR-2	0.446 951	0.455 81	0.449 163	0.492	1.458

From the change in irradiance column in Table 11.2, it is seen that the target pixel irradiance increased when the in-FOV adjacency effect was added. The percentage error column shows that the OSPA calculated values closely match the expected irradiance values when the in-FOV adjacency effect is added.

Table 11.3 illustrates the in-FOV adjacency effect on a 55% reflective target adjacent to a 3.5% reflective target.

From the change in irradiance column in Table 11.3, it is seen that the target pixel irradiance decreased when the in-FOV adjacency effect was added. The percentage error column shows that the OSPA calculated values closely match the expected irradiance values when the in-FOV adjacency effect is added.

When comparing the percentage change in irradiance between Tables 11.2 and 11.3, it is seen that the darker pixels (3.5% reflective) showed a greater change in irradiance than the brighter pixels (55% reflective).

Table 11.3: In-FOV adjacency effect for a 55% reflective target adjacent to a 3.5% reflective target.

Spectral band	55% reflective target pixel irradiance ($\times 10^{-13}$)(W/m ²)			Change in irradiance (%)	Error (%)
	OSSIM (no in-FOV adjacency)	Expected (with in-FOV adjacency)	OSPA (with in-FOV adjacency)		
Panchromatic	16.619 43	16.579 55	16.579 51	-0.241	0.0002
Coastal blue	2.973 128	2.967 40	2.971 475	-0.056	-0.137
Blue	4.187 566	4.179 15	4.185 168	-0.057	-0.144
Green	5.043 543	5.033 06	5.040 599	-0.058	-0.15
Yellow	2.891 941	2.885 94	2.890 223	-0.059	-0.148
Red	4.483 044	4.473 09	4.480 291	-0.061	-0.161
Red edge	2.442 452	2.438 65	2.441 352	-0.045	-0.111
NIR-1	6.272 256	6.257 97	6.268 203	-0.065	-0.164
NIR-2	3.374 387	3.366 53	3.372 212	-0.064	-0.169

11.2.3 Analysis of results

The MTF plot in Figure 11.3 confirmed that the in-FOV adjacency effect due to aerosol scattering had a negligible effect on the system MTF and that the unresolved pixels in Figure 11.2 were due to the detector cut-off frequency.

The results in Tables 11.2 and 11.3 showed that, as expected, the in-FOV adjacency effect reduced the image contrast by causing the brighter pixels to appear darker and the darker pixels to appear lighter. This can be seen by the increase in irradiance between the OSSIM and OSPA values in Table 11.2, and a decrease in irradiance between the OSSIM and OSPA values in Table 11.3. The results also showed that the brighter pixels were affected less by the adjacency effect than what the darker pixels were. This is seen by the small percentage of change in the at-sensor irradiance values in Table 11.3 and the larger percentage of change in the at-sensor irradiance values in Table 11.2.

From the percentage error values in Tables 11.2 and 11.3, it is seen that the OSPA calculated pixel irradiance values closely matched the expected pixel irradiance values. The NIR-2 irradiance value for the 3.5% reflective target had the largest mismatch of 1.458% from the expected value. The panchromatic irradiance value for the 3.5% reflective target had the smallest mismatch of -0.0002% from the expected value. This confirmed that the in-FOV adjacency effect had been calculated correctly in the OSPA environment.

It was concluded that the in-FOV adjacency effect term had been added correctly to the signature rendering equation, and that the in-FOV adjacency effect term was a good model of the in-FOV adjacency effect observed in satellite images.

11.3 Staggered detector focal plane layout

11.3.1 Test description

This test validates the staggered detector CCD array configuration as described in section 6.3.4. A configurable CCD array was implemented, in order to simulate the staggered focal plane array layout used in satellite sensors. The functionality allows detectors to be grouped into DSAs that are staggered in the focal plane by a specified distance. This was expected to result in the target appearing broken in the sensor image.

The focal plane layout was tested by scanning over a solid rectangular line target, Figure 11.1c. The target was positioned perpendicular to the sensor scan direction and was wider than the sensor across-track FOV, to ensure that each detector would image the target. The target pixel displacement in the sensor image was expected to be equivalent to the DSA stagger distance in the focal plane. The distortion of the straight line target geometry in the sensor output image made it easy to identify the number of DSAs and calculate the stagger distances.

The WorldView-2 focal plane array layout, Figure 6.4 in section 6.3.4, was used to test the configurable CCD array functionality. For this test the CCD arrays were set to a staggered row of 600 detectors for the panchromatic band and 150 detectors for each of the multispectral bands. The panchromatic CCD array was divided into 50 DSAs and each multispectral CCD array was divided into 10 DSAs. The simulation was set to stagger the odd numbered DSAs in the focal plane by 0.512 mm (64 panchromatic pixels) for the panchromatic band CCD array and 0.768 mm (24 multispectral pixels) for the multispectral band CCD arrays. The CCD arrays were not shifted in the focal plane. The panchromatic imaging CCD band had a pixel pitch of 8 μm and the multispectral band CCD arrays had a pixel pitch of 32 μm .

The satellite was set to scan the target using a descending orbit. According to the OSSIM observer orientation, the sensor's focal plane lies in the y-z plane of the body's coordinate system, where positive z points downward and x points along the track, as discussed in Appendix A.6. The satellite was pitched to point downward which would result in the positive z body coordinate axis pointing along the track in the opposite direction to the satellite forward motion. A positive stagger distance along the focal plane z-axis would, therefore, result in the DSAs shifting backwards along the track. The satellite was rolled to face the direction of motion for a descending orbit. The scan direction and DSA stagger configuration was expected to result in the even numbered DSAs imaging the target before the odd numbered DSAs.

All noise sources, such as detector non-uniformity and $NE\Delta\rho$, were excluded in this test so that the stagger distances and DSA number and size could easily be calculated. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.3.2 Results

All simulated WorldView-2 spectral imaging bands imaged the rectangular target and the results are listed in Table 11.4.

Table 11.4: Calculated staggered detector focal plane layout pixel values for each of the simulated WorldView-2 imaging spectral bands.

Spectral band	Number of detector subarrays (unitless)	Number of pixels in a detector subarray (unitless)	Number of detectors in the detector array (unitless)	Stagger pixel distance along the track (unitless)	Stagger distance along the track (mm)
Panchromatic	50	12	600	64	0.512
Coastal blue	10	15	150	24	0.768
Blue	10	15	150	24	0.768
Green	10	15	150	24	0.768
Yellow	10	15	150	24	0.768
Red	10	15	150	24	0.768
Red edge	10	15	150	24	0.768
NIR-1	10	15	150	24	0.768
NIR-2	10	15	150	24	0.768

The coastal blue imaging band results and calculations are discussed to illustrate how the values in Table 11.4 were calculated. Figure 11.4 illustrates the coastal blue imaging band results. The sensor output image is shown at the top left corner of the figure, a plot of the across-track DN values for the image row 18 is in the lower left of the figure and a plot of the along-track DN values for the image columns 59 and 60 is in the top right of the figure. The first pixel row in the sensor output image is the first along-track line scanned by the sensor.

The image in Figure 11.4 clearly shows geometric distortion in the target image. The straight rectangular target appears to be divided into two rows of five blocks each. Each rectangular block corresponds to one of the 10 DSAs in the CCD array. The target image first appears in row 18 of the image, in the columns corresponding to the even numbered DSAs. This indicates that the even numbered DSAs image the target before the odd numbered DSAs.

The DN values of image row 18 are plotted in order to confirm the number of detectors in a DSA. Markers are placed at the first and last pixels of one group of pixels corresponding to an even numbered DSA. The markers are placed at pixels number 45 and 59, which indicates that there are 15 detectors in the DSA.

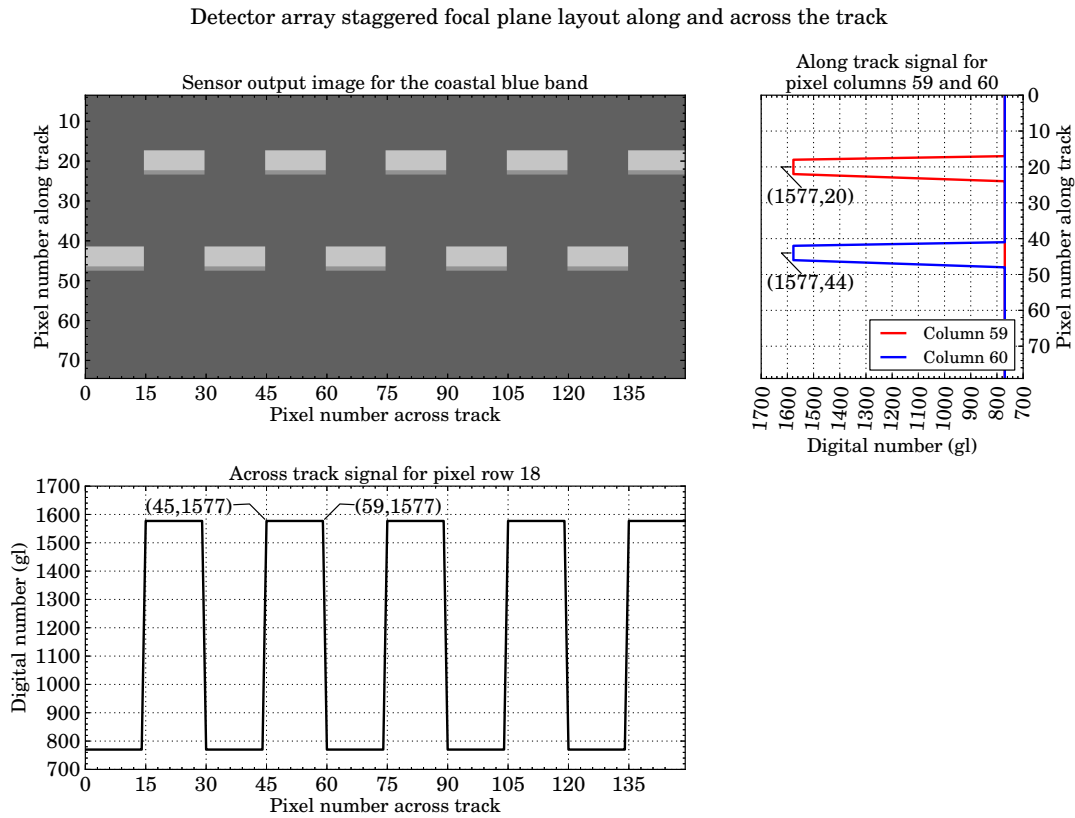


Figure 11.4: Sensor images and digital number values for a staggered detector focal plane layout for the simulated WorldView-2 satellite’s coastal blue multispectral band.

The total number of detectors in the CCD array is calculated, using the number of DSAs and the number of pixels in each DSA, to be $10 \times 15 = 150$.

The DN values of image columns 59 and 60 are plotted in order to calculate the number of pixels by which the DSAs have been staggered. The plot of image column 59 gives the DN values produced by a pixel in an even numbered DSA, while that of column 60 gives the DN values produced by a pixel in an odd numbered DSA. The two points that correspond to the same along-track location in columns 59 and 60 are marked. The pixel distance between these two points is equal to the number of pixels by which the odd DSAs have been staggered from the even DSAs. The stagger distance is calculated to be $44 - 20 = 24$ pixels, which is equal to a focal plane distance of $24 \times 32 \mu\text{m} = 0.768 \text{ mm}$.

11.3.3 Analysis of results

The results listed in Table 11.4 confirm that the expected results were achieved for all the spectral imaging bands. The results in the table are supported by the sensor output image and DN value plots in Figure 11.4. The expected geometric distortion in the image of the target is clearly visible in the sensor output image. The expected number of 10 DSAs and 150 detectors in the coastal blue band CCD array is also visible in the sensor

output image. This image confirms that, as expected, the even numbered DSAs imaged the target first. The plots of the DN values confirmed these results, as well as showing that the DSAs are staggered by the desired 0.768 mm distance.

It was concluded that the OSPA simulation environment was able to model a staggered detector array configuration correctly.

11.4 Shifted detector focal plane layout

11.4.1 Test description

This test validates the shifted detector CCD array configuration as described in section 6.3.4. A configurable CCD array was implemented in order to simulate the detector array shift in the focal plane array layout used in satellite sensors. The functionality allows a full detector array of an imaging band to be shifted in the focal plane by a specified distance. This was expected to result in each of the spectral band arrays imaging the target at a different time in the simulation. This would result in the imaged scene appearing to shift along the scan direction from one spectral band image to the next.

The shifted CCD array layout was tested by scanning over a solid rectangular line target, Figure 11.1c. The target was positioned perpendicular to the sensor scan direction and was wider than the sensor across-track FOV, to ensure that each detector would image the target. Each CCD array was shifted along the focal plane by a different distance relative to the unshifted zero position. The distance by which the target appeared to shift in the sensor output image is equivalent to the distance by which the CCD arrays had been shifted in the focal plane.

Similar to the test scenario in section 11.3, the WorldView-2 focal plane array layout was used to test the configurable CCD array functionality. The CCD array size was set to 600 detectors for the panchromatic band and 150 detectors for each of the multispectral bands. The panchromatic imaging CCD band had a pixel pitch of 8 μm and the multispectral band CCD arrays had a pixel pitch of 32 μm . The detector array was shifted along the positive z -axis of the focal plane and was not staggered. The NIR 1 CCD array was considered as the starting unshifted location in the focal plane and all imaging band CCD arrays were shifted relative to this point. The CCD array shift distances were set such that the multispectral band CCD arrays were separated by 0.096 mm (3 multispectral pixels) from one another and that the panchromatic band CCD array and the adjacent multispectral band CCD arrays were separated by 0.288 mm (9 multispectral pixels).

The satellite was set to scan the target in a descending orbit. The scan direction and WorldView-2 CCD array shift configuration was expected to result in the NIR 1 spectral band CCD array imaging the target first and the NIR 2 spectral band CCD array imaging the target last [63].

All noise sources, such as detector non-uniformity and $NE\Delta\rho$, were excluded in this test so that the exact shift distances could easily be calculated. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.4.2 Results

Figure 11.5 shows the output sensor CCD array images.

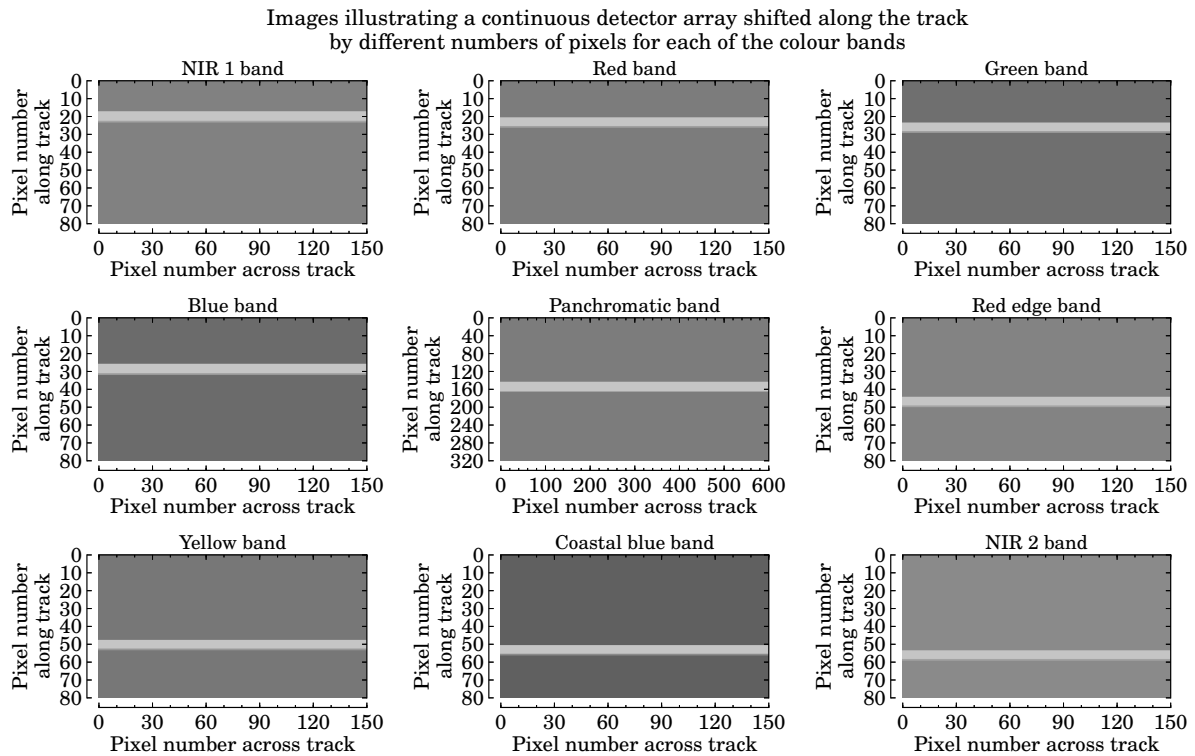


Figure 11.5: Simulated WorldView-2 satellite sensor output images for an along-track shifted detector array configuration.

As seen in Figure 11.5 the line target in the images appears to shift further down the image rows from one spectral band output image to the next, starting at the NIR 1 band output and ending at the NIR 2 band output. Note that the difference in detector size between the panchromatic and multispectral CCD arrays results in one multispectral image pixel being equivalent to 4×4 panchromatic pixels.

The DN values of the second image pixel column in each sensor output image in Figure 11.5 are plotted in Figure 11.6 against the image row position. The pixel row in which the target first appears in the image is marked in Figure 11.6. The panchromatic image pixel along-track count is plotted as the equivalent multispectral image pixel count.

Image pixel row 18 is the first image row in which the target appears in the unshifted NIR 1 band output and this is therefore considered to be the zero pixel shift point. The shift distance for a CCD array is calculated as the pixel difference between the zero shift

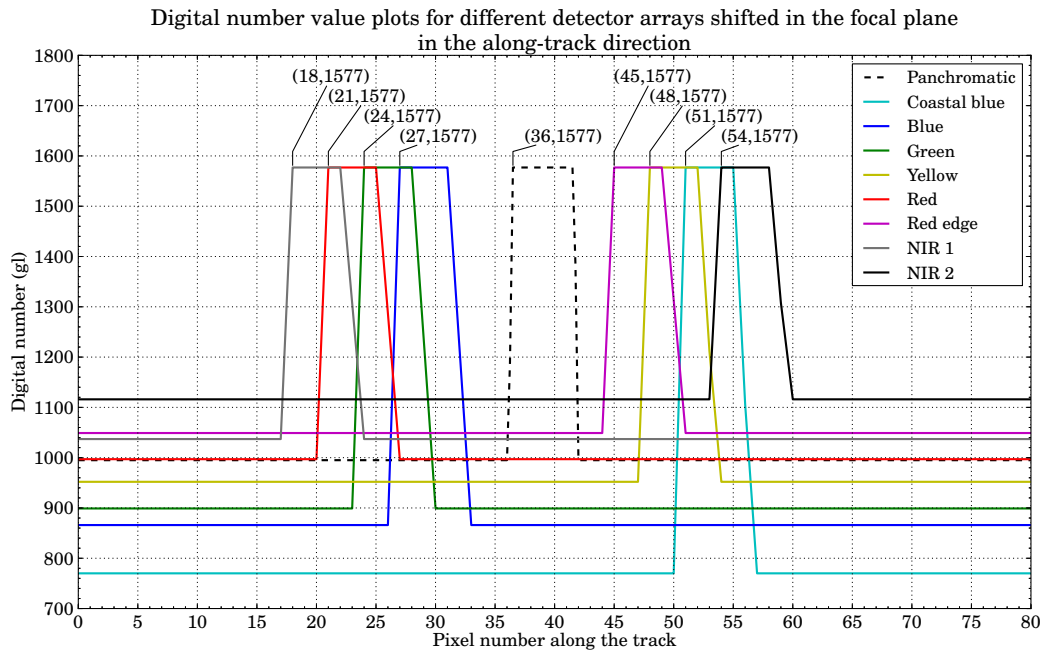


Figure 11.6: Simulated WorldView-2 satellite sensor digital number output for the first detector in each along-track shifted detector array, as the sensor scans along the track.

point and the image pixel row in which the target first appears in the output image for that band. For example, the panchromatic band target image first appears in pixel row 36. The pixel shift distance is calculated as being $36 - 18 = 18$ multispectral pixels along the track, which translates into a shift distance of $18 \times 32 \mu\text{m} = 0.576 \text{ mm}$. The shift distances for all imaging bands are listed in Table 11.5.

Table 11.5: Calculated shifted detector focal plane layout pixel values for each of the simulated WorldView-2 imaging spectral bands.

Spectral band	Expected shift distance along the track (mm)	Number of pixels shifted along the track (unitless)	Shift distance along the track (mm)
Panchromatic	0.576	18	0.576
Coastal blue	1.056	33	1.056
Blue	0.288	9	0.288
Green	0.192	6	0.192
Yellow	0.960	30	0.960
Red	0.096	3	0.096
Red edge	0.864	27	0.864
NIR-1	0.000	0	0.000
NIR-2	1.152	36	1.152

11.4.3 Analysis of results

As seen from Figures 11.5 and 11.6 and the results listed in Table 11.5, the desired shifted CCD array configuration was achieved for each of the imaging bands simulated.

The shifted array layout was observed to cause the target image to appear, as expected, to shift along the track. The shifted target location in Figures 11.5 and 11.6 confirmed that the order in which the CCD arrays image the scene matches the order in which the WorldView-2 imaging bands are placed in the focal plane, as shown in Figure 6.5 in section 6.3.4. The results listed in Table 11.5 further confirmed the shift order and also confirmed that the expected shift distance for each of the imaging bands was achieved.

It was concluded that the OSPA simulation environment was able to model a shifted CCD array focal plane layout correctly.

11.5 Noise equivalent reflectance

11.5.1 Test description

This test is used to confirm that the correct detector noise signal is observed in the output signal, for a given $NE\Delta\rho$ value. OSPA includes a model to simulate detector noise specified by the $NE\Delta\rho$ value, as described in section 6.3.6. The detector noise is tested by evaluating the SNR; if the detector noise is calculated correctly, then the output signal will produce the correct SNR.

For this test each of the sensor CCD imaging bands took an at-nadir snapshot of a striped line target, Figure 11.1c, of known reflectance. The panchromatic CCD array consisted of a row of 600 detectors and each multispectral CCD array consisted of a row of 150 detectors. The CCD arrays were not staggered or shifted.

The SNR for each of the imaging bands was calculated as shown in Listing 11.2.

```
#--- Calculate the difference in the high and low reflectance bar signal values
signal = (aveHighSignal - aveLowSignal)

#--- Calculate the average standard deviation of the output signal
noise = (stdDeviationHighSignal + stdDeviationstdLowSignal)/2.0

#--- Calculate the signal to noise ratio
SNR = signal/noise

#--- Confirm the noise equivalent reflectance value
nedp = noise/(maxDNofQuantisationRange - minDNofQuantisationRange)
```

Listing 11.2: Python code snippet used to calculate the signal to noise ratio for a digital number output signal produced when imaging a striped line target.

The signal is calculated as the difference in the reflectance between the high and low reflectance bars in the target. The noise is calculated as the average standard deviation of the DN output signal.

If the signal is set equivalent to the noise given by the $NE\Delta\rho$ value, then the sensor output signal is expected to have a SNR of one. A SNR of one is not a practical value with which to validate the detector noise model, since the signal is not distinguishable from the noise. A SNR of ten was chosen instead for this test. In order for the output signal to have a SNR of ten, the difference in reflectance between the target high and low reflectance bars should be ten times the $NE\Delta\rho$ value for that sensor imaging band. The high reflectance bars in the target were set to have a reflectance of 50%. The low reflectance bars in the target were set to have a reflectance of $0.5 - (10 \times 0.0021) = 0.479 = 47.9\%$ for the panchromatic imaging band, $0.5 - (10 \times 0.0015) = 0.485 = 48.5\%$ for the red and red edge spectral imaging bands and $0.5 - (10 \times 0.0016) = 0.484 = 48.4\%$ for the remaining WorldView-2 spectral bands simulated.

Only the detector noise due to the $NE\Delta\rho$ was considered and no other interference, such as the detector non-uniformity, was present in this test. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.5.2 Results

All simulated WorldView-2 spectral imaging bands imaged the striped line target and the results are listed in Table 11.6.

Table 11.6: Calculated signal to noise ratio and noise equivalent reflectance values for each of the simulated WorldView-2 imaging spectral bands. The expected signal to noise ratio was set to ten.

Spectral band	Signal (gl)	Noise (gl)	Signal to noise ratio (unitless)	Noise equivalent reflectance (unitless)	Error in signal to noise ratio (%)
Panchromatic	26.2265	2.619 10	10.0136	0.0021	-0.136
Coastal blue	19.2778	1.923 45	10.0225	0.0016	-0.225
Blue	19.2778	1.923 45	10.0225	0.0016	-0.225
Green	19.2778	1.923 45	10.0225	0.0016	-0.225
Yellow	19.2778	1.929 35	9.9919	0.0016	0.081
Red	18.1111	1.812 15	9.9943	0.0015	0.057
Red edge	18.0556	1.806 55	9.9945	0.0015	0.055
NIR-1	19.2778	1.923 45	10.0225	0.0016	-0.225
NIR-2	19.2778	1.923 45	10.0225	0.0016	-0.225

To illustrate how the values in Table 11.6 were calculated, the results and calculations for the panchromatic imaging band are discussed in detail. Figure 11.7 shows the plot of the panchromatic band DN output signal when a striped target of 50% and 47.9% reflectance is imaged.

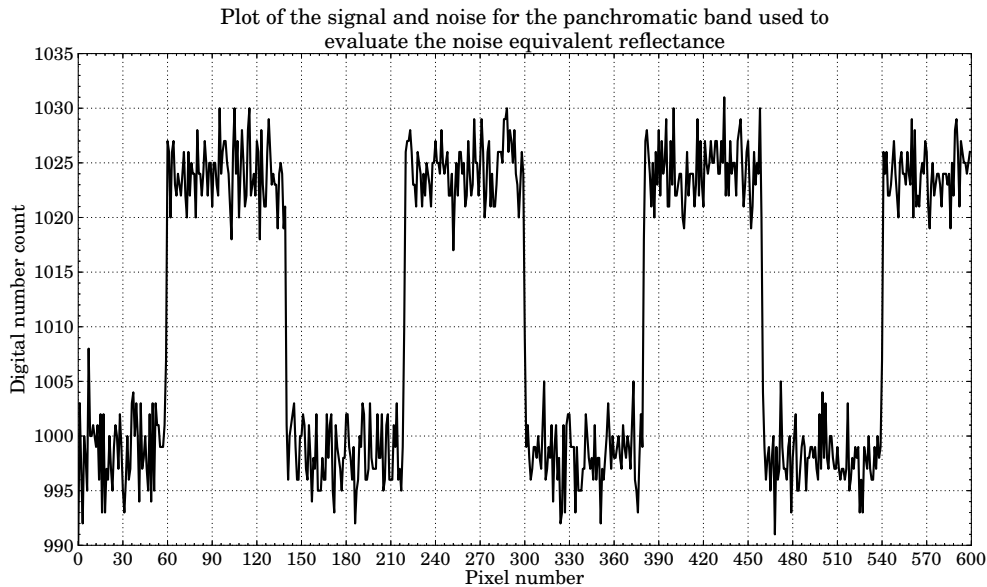


Figure 11.7: Plot of the OSPA simulation signal and noise output for the WorldView-2 satellite panchromatic band image of a striped line target with a reflectance of 50% and 47.9%. The target reflectance is set to produce a signal to noise ratio of 10.

Using the signal plotted in Figure 11.7 and the Python code in Listing 11.2, the SNR and $NE\Delta\rho$ values for the panchromatic imaging band are calculated. The average high reflectance bar signal was calculated to be 1024.3419 gl and the average low reflectance bar signal was calculated to be 998.1154 gl. The image band signal was then calculated as $1024.3419 - 998.1154 = 26.2265$ gl. The standard deviation of the high and low reflectance bar signals were calculated as 2.6349 gl and 2.6033 gl respectively. The average of the two noise values was taken, to give the image band noise of 2.6191 gl. The SNR was calculated to be $26.2265/2.6191 = 10.0136$. The $NE\Delta\rho$ was calculated by dividing the noise by the full quantisation range value $2.6191/(1638 - 410) = 0.0021$.

11.5.3 Analysis of results

The results in Table 11.6 showed that all the simulated imaging bands had an output signal with a SNR close to the desired SNR of ten. The percentage error in all the SNRs was less than 0.3%. When comparing the $NE\Delta\rho$ values in Table 11.6 to those in Table 6.1 in Chapter 6.3.6, it was seen that all imaging bands produced the expected $NE\Delta\rho$ values.

From the results in Table 11.6 it was concluded that the detector noise, given the $NE\Delta\rho$ value, was modelled accurately in the OSPA environment.

11.6 Analogue-to-digital conversion

11.6.1 Test description

This test validates the OSPA ADC model. An ADC model was added to the OSPA environment to perform data quantisation on the sensor output signal, as discussed in section 6.3.8. To evaluate the ADC algorithm, a target of known reflectance was imaged by each sensor spectral band. The sensor DN output was then analysed to confirm that the correct DN was calculated for the specified target reflectance.

For this test each of the sensor CCD imaging bands took an at-nadir snapshot of the greyscale target in Figure 11.1b. The greyscale target allowed the ADC algorithm to be tested for a range of reflectances from 5% to 95%. The detector array was positioned above the target so that the first detector in the array imaged the 5% reflective greyscale block and the last detector in the array imaged the 95% reflective greyscale block. To image the greyscale target, the panchromatic CCD array was set to a row of 432 detectors, and each multispectral CCD array to a row of 108 detectors. The detectors were aligned across the track and were not staggered or shifted.

To test the performance of the ADC algorithm at the lower and upper limits of the quantisation range, the sensor CCD imaging bands were set to take an at-nadir snapshot of a 0% and 100% reflective tarpaulin block. The 0% reflective target block was used to test the lower limit of the quantisation range and the 100% reflective target block to test the upper limit. These two targets were square blocks of uniform reflectance, similar to each reflective block in the greyscale bar target. Therefore, only a single detector element was needed to measure each of the 0% and 100% reflective target blocks.

If the expected DN output of 410 gl was produced when imaging the 0% reflective target, this confirmed that the flux and greylevel offset in the ADC algorithm had been implemented correctly. If the expected DN output of 1638 gl was produced when imaging the 100% reflective target then it was confirmed that the upper limit of the quantisation range had been set correctly in the OSPA ADC algorithm. If the expected DN output was produced when imaging the different reflective blocks in the greyscale target, it was confirmed that the flux and greylevel gain in the ADC algorithm had been implemented correctly.

All noise sources such as detector non-uniformity and $NE\Delta\rho$ were excluded in this test. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.6.2 Results

All the WorldView-2 spectral bands simulated imaged the greyscale bar target and the 0% and 100% reflective target blocks. The DN output results are listed in Table 11.7.

Table 11.7: Calculated digital numbers for a range of target reflectance values for each of the WorldView-2 imaging spectral bands simulated.

Target reflectance (unitless)	Expected digital number (gl)	Imaging band digital number (gl)								
		Panchromatic	Coastal blue	Blue	Green	Yellow	Red	Red edge	NIR-1	NIR-2
0.00	410	410	410	410	410	410	410	410	410	410
0.05	471	471	471	471	471	471	471	471	471	471
0.15	594	594	594	594	594	594	594	594	594	594
0.25	717	717	717	717	717	717	717	717	717	717
0.35	840	840	840	840	840	840	840	840	840	840
0.45	963	963	963	963	963	963	963	963	963	963
0.55	1085	1086	1085	1085	1085	1085	1085	1086	1085	1085
0.65	1208	1208	1208	1208	1208	1208	1208	1208	1208	1208
0.75	1331	1331	1331	1331	1331	1331	1331	1331	1331	1331
0.85	1454	1454	1454	1454	1454	1454	1454	1454	1454	1454
0.95	1577	1577	1577	1577	1577	1577	1577	1577	1577	1577
1.00	1638	1638	1638	1638	1638	1638	1638	1638	1638	1638

11.6.3 Analysis of results

As seen in Table 11.7, each of the imaging bands produced the expected DN value for the specific target reflectance. Only the DNs for the 0.55 reflective target, imaged by the panchromatic and red edge imaging bands, produced a DN 1 gl greater than expected. This error was the result of rounding off and is less than 1%. The results for the maximum and minimum target reflectance values of 0 and 1, indicated that the ADC correctly calculated the expected upper and lower limits of the quantisation range. All imaging bands produced a set of DN numbers, between the desired quantisation range of 410 gl to 1638 gl, that correctly represented the target reflectance. These results confirmed that the flux and greylevel offset and gain values had been set up and included in the ADC algorithm correctly.

The test results indicated that the pixel DN equation, as shown in (6.3.6), is an accurate model of the ADC. The simulation results indicate that this equation had been correctly implemented in the OSPA simulation environment.

11.7 Across-track scanning using a rotating scan mirror

11.7.1 Test description

This test is used to validate the across-track scanning motion of the rotating scan mirror functionality, as discussed in section 7.4. The rotating scan mirror functionality was included in OSPA in order to simulate the scanning motion of paddlebroom type scanners as described in section 2.2.1.4. This type of across-track scanning was expected to result in no geometric distortion in the sensor output image, provided the satellite attitude and altitude remained constant while scanning over the target.

To test the rotating scanning mirror algorithm the sensor was set to image a triangular target, similar to Figure 11.1d, where the low reflectance material is on the right half of the triangle. The rotating scan mirror across-track scanner functionality was validated with the assumption that the satellite advanced along the track only once the scan mirror had scanned across the full swath width and no scan lines overlapped. The satellite altitude was also set to remain unchanged for the simulation run. The image was scanned using a CCD array that was aligned along the track. The detectors formed a single column of 16 detectors for the panchromatic imaging band and a column of four detectors for each multispectral band. The detector arrays had not been staggered or shifted. The sensor scanned the scene using a descending orbit and the rotating scan mirror was set to scan across the track from left to right for each scan line. The sensor output image was, therefore, expected to show a downwards pointing triangle with the pixels corresponding to the representation of the low reflectance target material (dark pixels) as being on the left of the triangle image, as explained in section 11.9.

To simulate the rotating mirror scanning motion the gimbal yaw attitude was adjusted to tilt the sensor's line of sight across the track while the gimbal's roll and pitch attitudes remained constant throughout the simulation run. The gimbal was set to have an initial yaw attitude of -0.1203 mrad and a yaw increment angle of 0.6015 μ rad. The gimbal attitude is incremented every 62 μ s to scan an across-track FOV of 0.2406 mrad. For a rotating scan mirror, each swath scan is scanned in the same direction, therefore, the gimbal starting attitude was expected to be the same, -0.1203 mrad, for each swath scan, which would result in the gimbal yaw attitude plot against the simulation time being sawtoothed. The yaw attitude increment and simulation time increment values matched the WorldView-2 panchromatic imaging band detector IFOV and dwell time, respectively. The WorldView-2 multispectral bands will image the scene every fourth time increment, since the WorldView-2 multispectral imaging bands are set to have a dwell time four times greater than the panchromatic band. These values lead to the gimbal being incremented by 0.2406 mrad/ 0.6015 μ rad = 400 times in one swath scan, which results in the swath

width having 401 pixels for the panchromatic band and $401/4 = 100.25 \approx 100$ pixels for the multispectral bands.

The satellite along-track position was expected to be incremented every $401 \times 62 \times 10^{-6} \text{ s} = 0.0248 \text{ s}$, by 16 panchromatic pixels. For the panchromatic band GSD of 0.4632 m, a 16 panchromatic pixel distance would result in a 7.4112 m distance on the ground. The satellite attitude, altitude and across-track position remained constant for the simulation run.

All noise sources such as detector non-uniformity and $NE\Delta\rho$ were excluded in this test. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.7.2 Results

All simulated WorldView-2 imaging bands were used to image the target. For purposes of illustration and to confirm that the scan direction was correct, the output image produced by the panchromatic imaging band is shown in Figure 11.8.

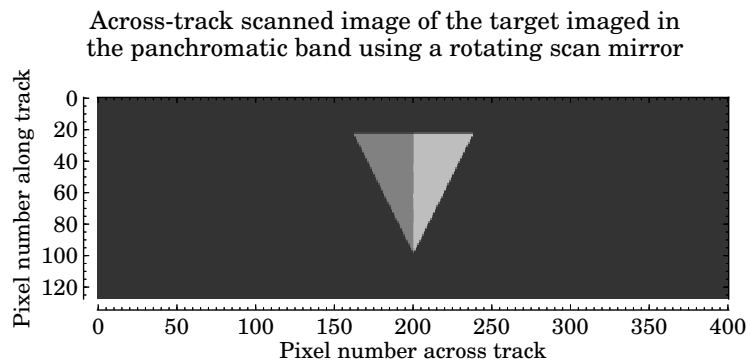


Figure 11.8: Image scanned across-track using a rotating scan mirror.

As seen in Figure 11.8, the triangular target points downwards with all the low reflectance (dark) target pixels on the left half of the triangular image. There is no scan line distortion in the image.

The gimbal roll, yaw and pitch attitude values are plotted against the simulation time in Figure 11.9. The plots are marked at the start and end points of a single across-track scan interval. The start of the next swath scan point has also been marked.

From the yaw angle plot in Figure 11.9, it can be seen that the plot has a sawtooth shape. From the markers it can be seen that the gimbal yaw attitude was set to -0.1203 mrad at the start of a single swath scan, the gimbal yaw attitude was then incremented to 0.1203 mrad. This resulted in an across-track FOV of 0.2406 mrad. The time taken to scan a full swath width was calculated to be $0.074524 - 0.049724 = 0.0248$ s. This plot also showed that the start of each swath scan had the same gimbal yaw attitude; this is

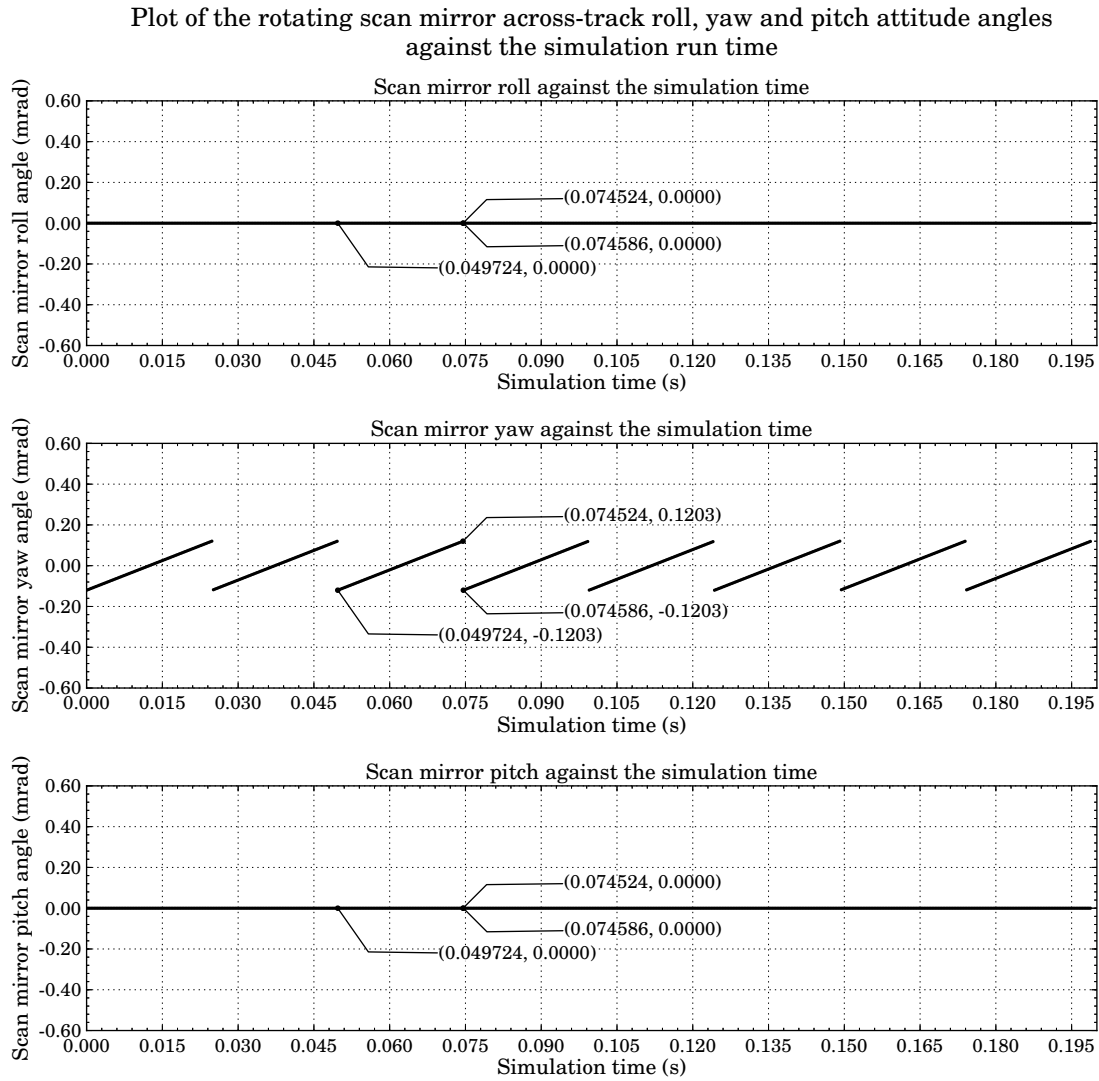


Figure 11.9: Rotating scan mirror attitude against simulation time.

seen by the attitude marked at the 0.049 72 s and 0.074 586 s simulation time. The gimbal yaw attitude was shown to be adjusted every $0.074586 - 0.074524 = 62 \mu\text{s}$, this led to $0.0248/62 \times 10^{-6} = 400$ increments in each swath width. The yaw attitude increment was calculated to be $(0.1203 \times 10^{-3} - (-0.1203 \times 10^{-3}))/400 = 0.6015 \mu\text{rad}$. The gimbal roll and pitch attitude angles remained constant throughout the simulation run.

The satellite along and across-track positions and altitude against the simulation time are plotted in Figure 11.10.

From the satellite trajectory plot in Figure 11.10, it is seen that the satellite moves along the track by $2.4849 - 9.8961 = -7.4112 \text{ m}$ each $0.074524 - 0.049724 = 0.0248 \text{ s}$ time interval. This time interval is equal to the time it takes for the sensor to scan across the full swath width. The satellite along-track position remains constant during that time interval. The satellite across-track position remains in the centre of the track for the full simulation run.

Plot of the satellite trajectory coordinates against the simulation run time

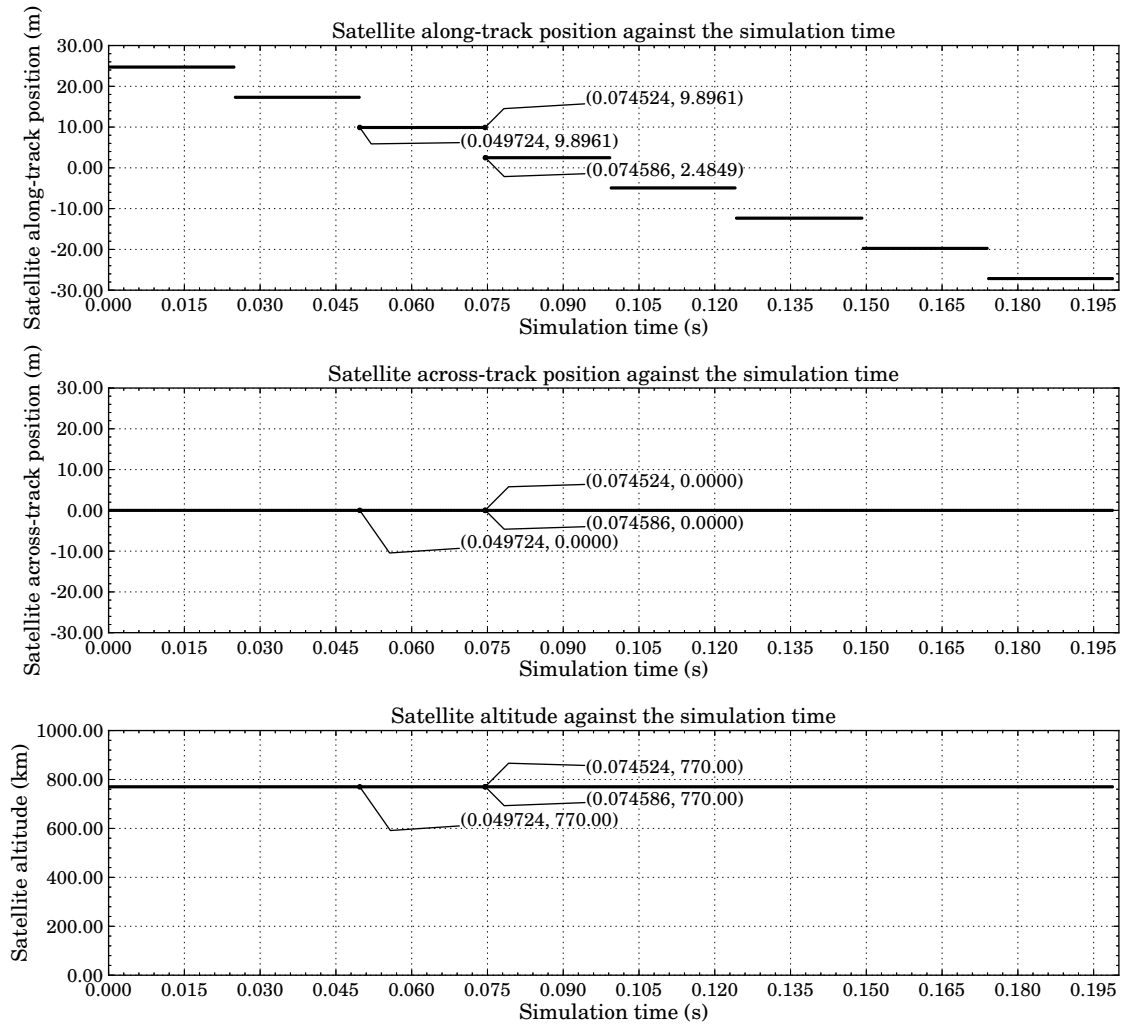


Figure 11.10: Satellite trajectory coordinates against simulation time while imaging the scene using a rotating scan mirror.

11.7.3 Analysis of results

From Figure 11.8 it can be seen that the image of the target matched the expected image geometry of a downward pointing triangle with all the darker target pixels appearing on the left side of the target image. This provided visual confirmation that successive scan lines were imaged in the same direction across the track. This finding is further supported by the sawtooth shaped gimbal yaw attitude plot in Figure 11.9. The plot showed that all scan lines had the same starting gimbal yaw attitude. The attitude plot further confirmed that the scanned swath width matched the expected swath FOV of 0.2406 mrad. The scanned swath width was shown to be symmetrical about the centre of the swath width. The yaw plot also showed that the gimbal attitude was incremented every 62 μ s by the expected increment amount of 0.6015 μ rad.

The simulation time of the markers at the beginning and end of one satellite along-track

position step in Figure 11.10 corresponds with the simulation time for the markers at the start and end of the corresponding swath scan in the yaw angle plot in Figure 11.9. This showed that the satellite advanced forward along the track only once the full swath width had been scanned. The desired satellite along-track motion was achieved, since the satellite has been shown to move by -7.4112 m once the full swath width had been scanned. This increment distance showed that the satellite followed the desired descending orbit.

From the plotted results and the sensor output image it was concluded that the OSPA rotating scan mirror operated as expected and was able to simulate the desired across-track rotating scanner accurately.

11.8 Across-track scanning using an oscillating scan mirror

11.8.1 Test description

This test is used to validate the across-track scanning motion of the oscillating scan mirror functionality, as discussed in section 7.4. The oscillating scan mirror functionality was included in OSPA in order to simulate the scanning motion of whiskbroom type scanners as described in section 2.2.1.4. This type of across-track scanning was expected to result in the east and west directions in the image appearing flipped in alternating across track scan lines. No other geometric distortion was expected in the sensor output image, provided that the satellite attitude and altitude remained constant while scanning over the target.

To test the oscillating scanning mirror algorithm, the sensor was set to image a triangular target, similar to that in Figure 11.1d, where the low reflectance material was on the right half of the triangle. The oscillating scan mirror across-track scanner functionality was validated with the assumption that the satellite advanced along the track only once the scan mirror had scanned across the full swath width and no scan lines overlapped. The satellite altitude was also set to remain unchanged for the simulation run. The image was scanned using a CCD array that was aligned along the track. The detectors formed a single column of 16 detectors for the panchromatic imaging band and a column of four detectors for each multispectral band. The detector arrays had not been staggered or shifted. The sensor scanned the scene using a descending orbit and the oscillating scan mirror was set to scan the first across-track line from left to right. The sensor output image was expected to show a downward pointing triangle, as explained in section 11.9. For a gimbal across-track scan direction from left to right, the high reflectance (light) pixels of the triangle target image were expected to appear on the right in the sensor image. For a scan direction from right to left, the light target pixels were expected to appear on the left in the image.

To simulate the oscillating scan mirror, the gimbal yaw angle was adjusted to first increment along the swath scan, in order to scan from left to right across the track. The next scan line would be the first across-track scan line to image the target and would be from right to left, so that the gimbal will decrement along the swath scan. This would result in a gimbal yaw attitude plot having a triangular shape. The gimbal roll and pitch angles would remain constant throughout the simulation run. The gimbal was set to have an initial yaw attitude of -0.1203 mrad, the gimbal attitude was then incremented by 0.6015 μ rad every 62 μ s, to scan a swath FOV of 0.2406 mrad. The gimbal starting attitude for the next scan line was expected to be equal to the end attitude of the previous scan line. It was, therefore, expected that the starting gimbal attitude for each swath scan would either be 0.1203 mrad or -0.1203 mrad. The yaw attitude increment and simulation time increment values matched the WorldView-2 panchromatic imaging band detector IFOV and dwell time, respectively. The WorldView-2 multispectral bands would image the scene every fourth gimbal yaw increment since the multispectral imaging bands had a dwell time four times greater than that of the panchromatic band. These values led to the gimbal being incremented 0.2406 mrad/ 0.6015 μ rad = 400 times. The swath width would, therefore, be 401 pixels wide for the panchromatic band and $401/4 = 100.25 \approx 100$ pixels wide for the multispectral bands.

The satellite along-track position was expected to be incremented every $401 \times 62 \times 10^{-6} = 0.0248$ s, by 16 panchromatic pixels. For the panchromatic band GSD of 0.4632 m, a 16 panchromatic pixel distance would result in a 7.4112 m distance on the ground. The satellite attitude, altitude and across-track position remained constant for the simulation run.

All noise sources, such as detector non-uniformity and $NE\Delta\rho$, were excluded in this test. The test was run using the good visibility Tropical atmospheric model, described in section 5.3.1.

11.8.2 Results

All simulated WorldView-2 imaging bands were used to image the target. For illustration purposes and to confirm that the scan direction was correct, the output image produced by the panchromatic imaging band is shown in Figure 11.11. As seen in Figure 11.11, the high reflectance (light) and low reflectance (dark) pixels in the downward pointing triangular target appeared on different sides of the target image for alternating scan lines. The high reflectance target pixels first appeared to be on the left of the target, and then on the right in the next image scan line.

The gimbal roll, yaw and pitch attitude values are plotted in Figure 11.12. The plots were marked at the beginning and end points of a single across-track scan interval. The end of the next swath scan point and the first gimbal attitude increment were also marked.

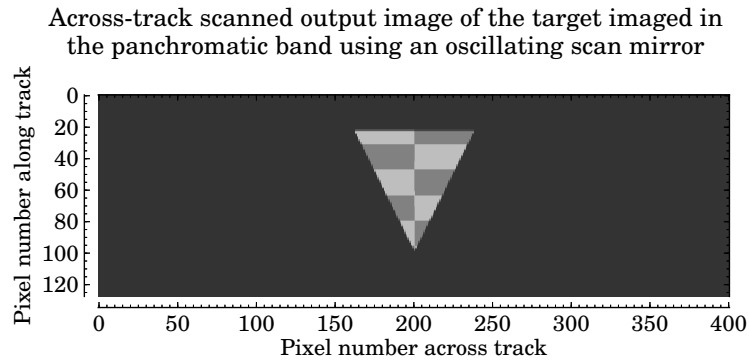


Figure 11.11: Image scanned across-track using an oscillating scan mirror.

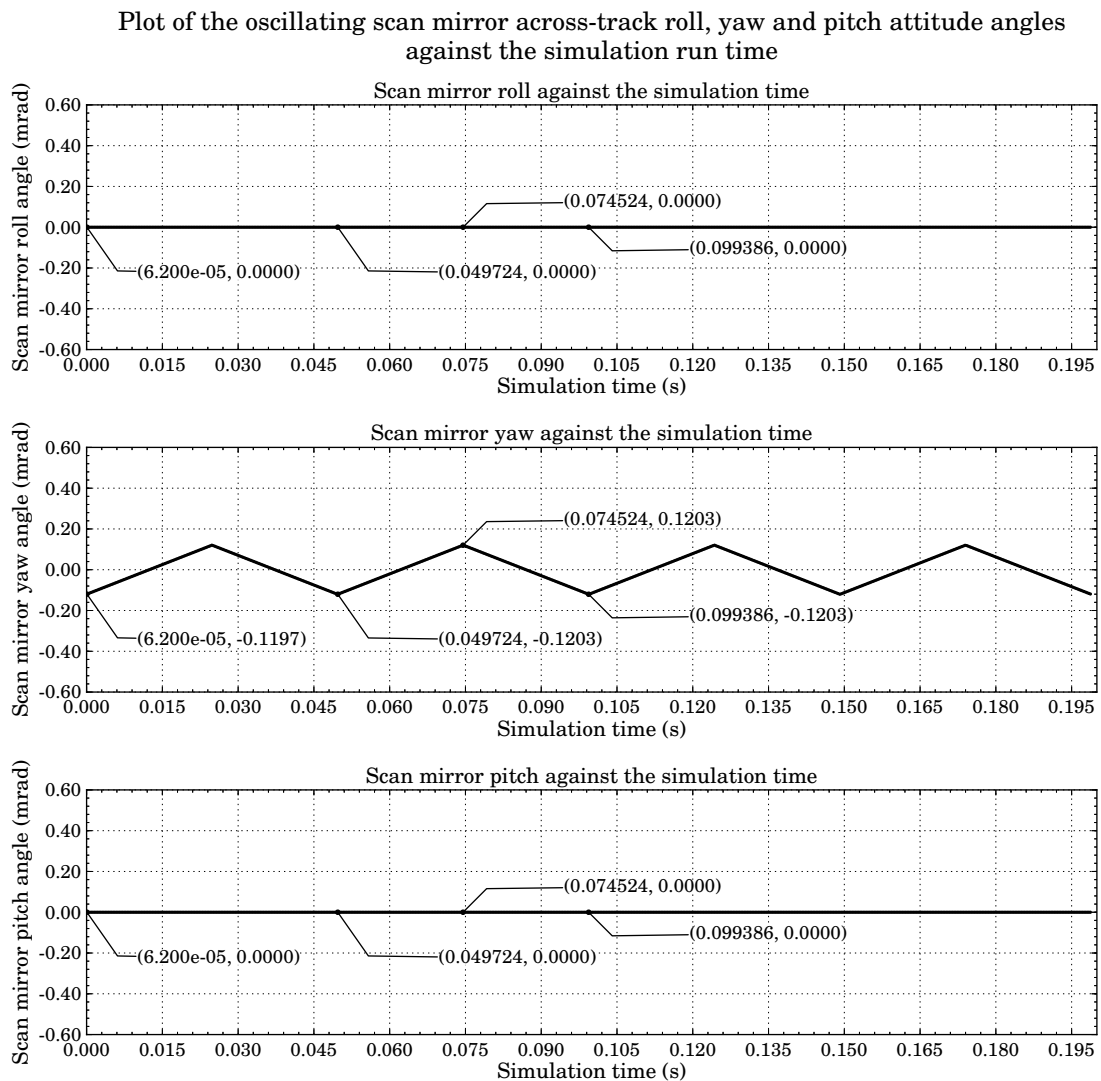


Figure 11.12: Oscillating scan mirror attitude against simulation time.

From the yaw angle plot in Figure 11.12, it is seen that the plot has a triangular shape. From the markers it is seen that the gimbal yaw attitude was set to -0.1203 mrad at the start of a single swath scan; the yaw attitude was then incremented to 0.1203 mrad.

This resulted in an across-track FOV of 0.2406 mrad. The time taken to scan a full swath-width was calculated to be $0.074524 - 0.049724 = 0.0248$ s. This plot also shows that the start of each swath scan had the same gimbal yaw attitude as the gimbal end attitude of the previous swath scan. This can be seen by the triangular shape of the graph, and the marker at 0.074 524s, which marks the end of the swath scan which started at marker 0.049 72s. The 0.074 524s marker also marks the start of the next swath scan that ends at the 0.099 386s marker. The marker placed at the first attitude increment showed that the attitude was adjusted every $62 \mu\text{s}$, this resulted in $0.0248/62 \times 10^{-6} = 400$ increments in each swath width. The yaw attitude increment was calculated to be $(0.1203 \times 10^{-3} - (-0.1203 \times 10^{-3}))/400 = 0.6015 \mu\text{rad}$ per time interval. The gimbal roll and pitch attitude angles remained constant throughout the simulation run.

The satellite's along and across-track positions and altitude are plotted in Figure 11.13.

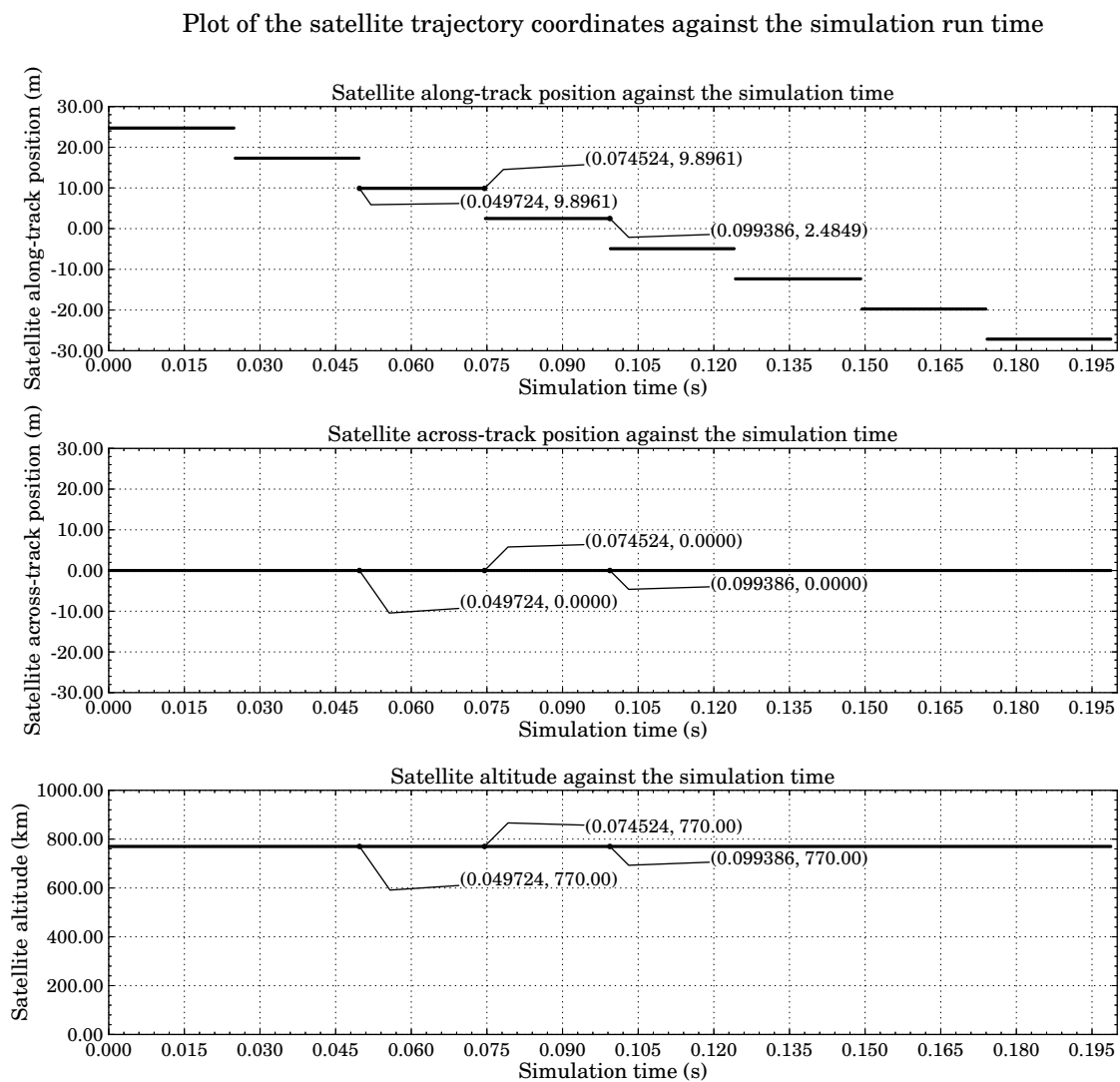


Figure 11.13: Satellite trajectory coordinates against simulation time while imaging the scene using an oscillating scan mirror.

From the satellite along-track position plot in Figure 11.13, it is seen that the satellite moves along the track by $2.4849 - 9.8961 = -7.4112$ m each $0.074524 - 0.049724 = 0.0248$ s time interval. The satellite's along-track position remains constant during that time interval. The satellite across-track position remains in the centre of the track for the full simulation run.

11.8.3 Analysis of results

In Figure 11.11 it can be seen that the image of the target matches the expected image geometry of a distorted downward pointing triangle. The target image clearly showed the distortion effect of the oscillating scan mirror, since the placement of the light and dark target pixels is dependent on the across-track scan direction. This indicated that successive scan lines are imaged in opposite directions across the track. It is also seen in the image that the first across-track scan to image the target causes the light target pixels to appear on the left of the image. This matched the expected results for the across-track scan direction from right to left.

The triangular shaped yaw attitude plot, in Figure 11.12, further supported the visual analysis that alternating swaths are scanned in opposite directions across the track. The yaw angle plot showed that the starting gimbal yaw attitude value for a swath scan was the same as the end gimbal yaw attitude value for the previous swath scan. The attitude plot further confirmed that the scanned swath width matched the expected across-track FOV of 0.2406 mrad. The scanned swath line was shown to be symmetrical about the centre of the swath. The yaw plot also showed that the gimbal attitude was incremented every 62 μ s by the expected increment amount of 0.6015 μ rad.

The desired satellite along-track motion, Figure 11.13, had been achieved, since the satellite had been shown to move by -7.4112 m only once the full swath width had been scanned. This increment showed that the satellite followed the desired descending orbit.

From the plotted results and the sensor output image it was concluded that the OSPA oscillating scan mirror operated as expected and was able to simulate the desired across-track oscillating mirror scanner accurately.

11.9 Image appearance due to orbit direction and earth rotation

11.9.1 Test description

The orbit direction and the rotation of the earth affect the appearance of the scene in an image. The northerly and southerly directions and easterly and westerly directions of a scene will appear to have been flipped, depending on whether the satellite is in an

ascending or descending orbit. The rotation of the earth will cause the scene to appear sheared in the image. This test is used to ensure that the OSPA sensor output image includes these effects.

To simulate the orbit direction and earth rotation effects on the output image, the satellite was set to scan over a triangular target on the surface of a spherical target, Figure 11.1e. The triangle pointed downwards and consisted of two different materials, a high reflective material (light pixels) was on the right half of the triangle and a low reflective material (dark pixels) was on the left half of the triangle. The satellite sensor was set to scan over the triangular target using an ascending or descending orbit while the sphere is stationary and while it rotates from west to east. The sensor images were visually evaluated to ensure that the target appeared as expected.

If an ascending orbit is used to scan the triangle target then the directions of north and south of the scene are expected to be flipped, so that the target appears to point upwards and the left side of the triangle appears darker than the right. If the spherical target is rotating while the sensor is imaging the triangular target, then the target is expected to appear sheared across the track, with the triangle base being shifted further to the right in the image, away from the triangle point. If a descending orbit is used to scan the triangle target then the target is expected to point downward with the directions of east and west of the scene being flipped, so that the target appears to be darker on the right. If the spherical target is rotating when the sensor is imaging the triangular target, then the target is expected to appear sheared across the track, with the triangle point being shifted further to the left in the image, away from the triangle base.

The along-track scanning sensor was set to follow a direct orbit path between two latitude and longitude points on the sphere. The trajectory path, detector specifications and orbit altitude were set so that the triangular target on the sphere remained within the sensor's FOV throughout the simulation run. It is necessary to simulate only one sensor spectral band, since the image appearance effects that were of interest in this test were not dependent on the spectral range in which the target is imaged.

The panchromatic imaging band CCD array was used in this test and was set to comprise 1500 detectors arranged in a single row across the track. The detector array was not staggered or shifted within the focal plane and no detector noise or non-uniformity were present. The detector pitch was set to $54.269\ \mu\text{m}$ and a focal length of 13.3 m and orbit altitude of 770 km were used to achieve a GSD of 3.141 89 m. The radius of the sphere target was set to be 25 512.4 m, which is smaller than that of the earth. The sphere target radius, in conjunction with the sensor settings, allowed the sensor to image a larger area of the sphere target with a small detector array, and in so doing decreased the simulation run time and prevented an out-of-memory failure.

The earth rotates at an angular velocity of $72.722\ \mu\text{rad/s}$, this rotation is not practical for the simulation since a long run time would be required to see the effect of the rotating

earth in the output image. For this test the spherical target was set to rotate at an angular velocity of 1.173 rad/s, which resulted in the target rotating by $72.722 \mu\text{rad}$ each dwell time interval of $62 \mu\text{s}$. This angular velocity will cause the target to appear to shift across the track by $72.722 \times 10^{-6} \times 25512.4 = 1.8553 \text{ m}$ each time interval. The expected gradient of the shift in the image was then calculated to be $3.14189/1.8553 = 1.693$ pixels for both ascending and descending orbits. The descending orbit was expected to have a negative gradient shift.

The atmosphere was not present in this test.

11.9.2 Results

The sensor output image results for the panchromatic spectral band CCD array imaging a downward pointing triangle on the surface of the sphere, are shown in Figure 11.14.

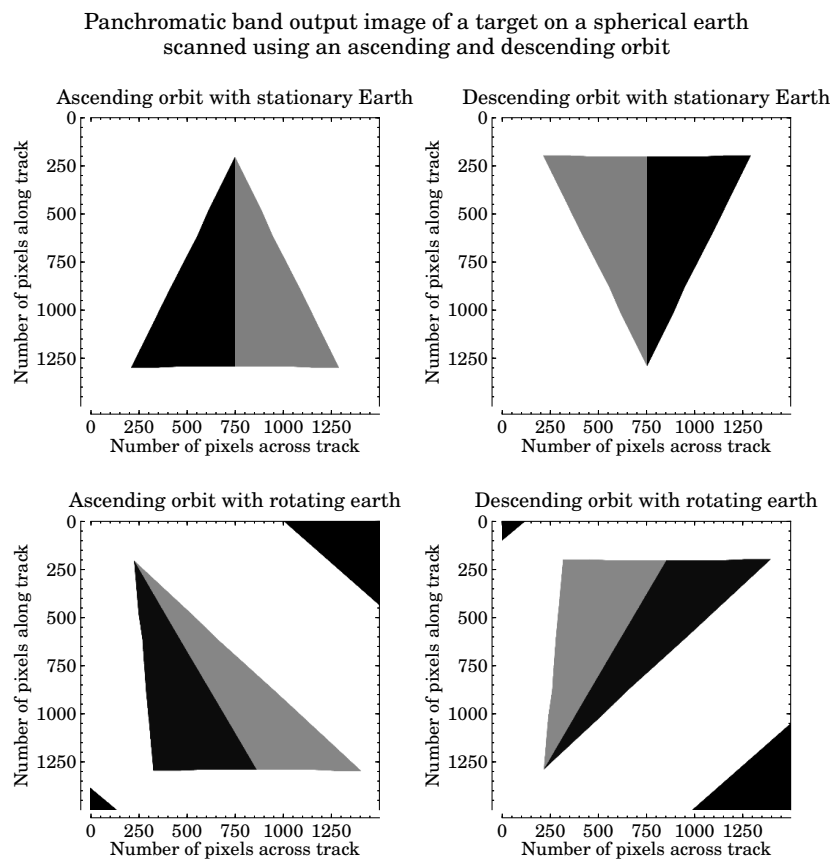


Figure 11.14: Test target scanned along the track using different orbit directions and either a stationary or a rotating earth.

Note that the top left corner of the images is the (0,0) pixel and this is the first pixel row scanned by the along-track (pushbroom) scanner.

Figure 11.14 shows that the triangular target appears to point upwards in the ascending orbit images, indicating that the directions of north and south of the scene are flipped in

the image. The triangular target appeared to be darker on the right side in the descending orbit images, indicating that the directions of east and west of the scene had been flipped in the image. The bottom two images clearly showed the distortion in the target geometry caused by the rotating earth, where the triangular target point appeared further to the left of the images than the target base.

Figure 11.15 plots the DN values for the image rows 250 and 1250, that intersect the triangular target, in order to calculate the shift distance seen in the scan lines. As seen in

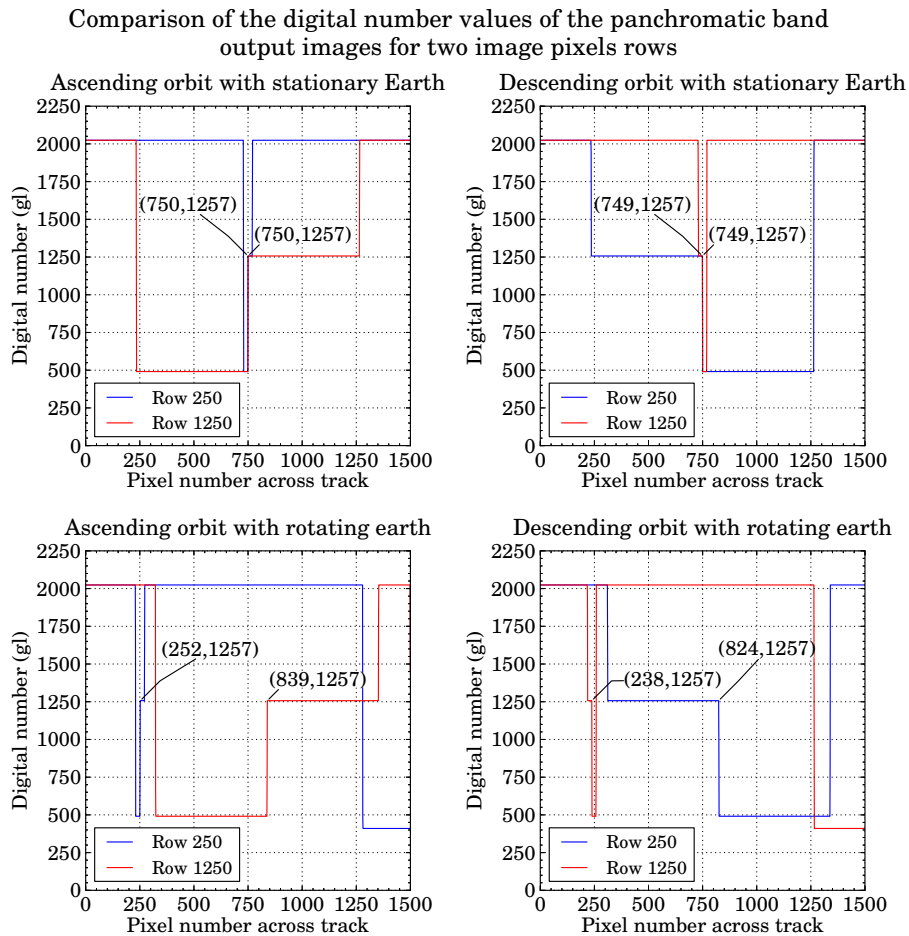


Figure 11.15: Digital number cross section values of a triangular target when imaged using different orbit directions and a stationary or rotating earth.

the plots for the stationary earth target, Figure 11.15, the pixel at the point of the triangle perfectly aligns along the track with the pixel at the centre of the triangle base. When the earth target rotates, it is seen that these two pixel points no longer align. The difference in pixel value was used to calculate the gradient of the shift in the scan lines. For the ascending orbit the shift gradient was calculated to be $(250 - 1250)/(252 - 839) = 1.704$ pixels, and for the descending orbit the shift was calculated to be $(250 - 1250)/(824 - 238) = -1.706$ pixels. The percentage difference between these gradients and the expected gradient

of 1.693 pixels was calculated to be $((1.693 - 1.704)/1.693) \times 100 = -0.649\%$ and $((-1.693 - (-1.706))/-1.693) \times 100 = -0.768\%$.

11.9.3 Analysis of results

The images in Figure 11.14 clearly showed that, as expected, the northerly and southerly directions and the easterly and westerly directions in the image had been flipped correctly, according to the orbit direction. The target in the image outputs appeared as expected, which indicated that the scene had been correctly scanned by the OSPA simulated sensor and the image output appearance is dependent on the orbit direction of the satellite.

The shear of the triangle in the images for a rotating earth matched the expected gradient of the shift in the image to within 1%. The across-track direction with which the scan lines appeared shifted matched the expected values. It was concluded that the shift in the scan lines due to the rotating earth had been correctly implemented.

From the sensor output images and plotted results it was concluded from the sensor image's appearance, that the orbit direction and earth rotation had been correctly implemented in the OSPA environment.

11.10 Satellite full circular orbit

11.10.1 Test description

This test point validates the full circular orbit trajectory model. OSPA provides the functionality to define the satellite's motion as a full circular orbit, as discussed in Chapter 8. To evaluate the full circular orbit trajectory, the calculated satellite position and attitude for the orbit path around a perfectly spherical earth target was plotted and analysed. The orbit trajectory plot was inspected to ensure that the orbit direction, trajectory shape, RAAN angle and inclination angle were correct. The orbit attitude plots were inspected to ensure that the satellite's attitude for a specific location in the orbit would always allow the satellite to point to the centre of the target and remain perpendicular to the satellite track.

In this test only the satellite trajectory was of interest, not the sensor output image. The orbit algorithm was set to calculate the satellite's trajectory and attitude for the WorldView-2 satellite orbit, for a single full circular orbit around the earth. The earth was modelled by a spherical target, Figure 11.1e, which had a radius of 6378.1 km and was assumed to be stationary. The orbit to be calculated was set to be a descending orbit with an inclination angle of 97.8° and a RAAN angle of 180° . The RAAN angle was chosen for convenience, so that the starting value of the orbit would be at 0° latitude and 0° longitude, which for this test was assumed to be the sun side of the earth. The orbit was set to have

an altitude of 770 km above the earth. The orbit was calculated relative to a reference point set to be at the top of the earth ($90^{\circ}0'0''\text{N}$, $0^{\circ}0'0''\text{E}$). The satellite attitude was set to be calculated for the yaw-pitch-roll Euler angle rotation order.

The calculated circular orbit ground track was expected to display longitudinal and equatorial symmetry. The satellite's yaw attitude angle was expected to be equivalent to the negative of the satellite's longitude position. This yaw angle would allow the satellite to point to the correct longitude location on the surface of the earth. The satellite's pitch attitude angle was expected to be equivalent to the negative of the satellite's latitude position. This pitch angle would allow the satellite to look down towards the centre of the earth. The satellite's roll attitude angle was expected to correspond to the orbit inclination angle and to take into account the direction of the orbit. This allowed the satellite's sensor to face forward along the track, as well as to be aligned perpendicularly to the track. The satellite's roll angle should remain constant throughout the trajectory.

11.10.2 Results

The results of the calculated satellite latitude and longitude trajectory path, as the satellite orbits the earth, is plotted in Figure 11.16. The ascending and descending nodes and the maximum latitude point are marked on the plot. Each marker provides the latitude, longitude and orbit altitude value at that point in the trajectory.

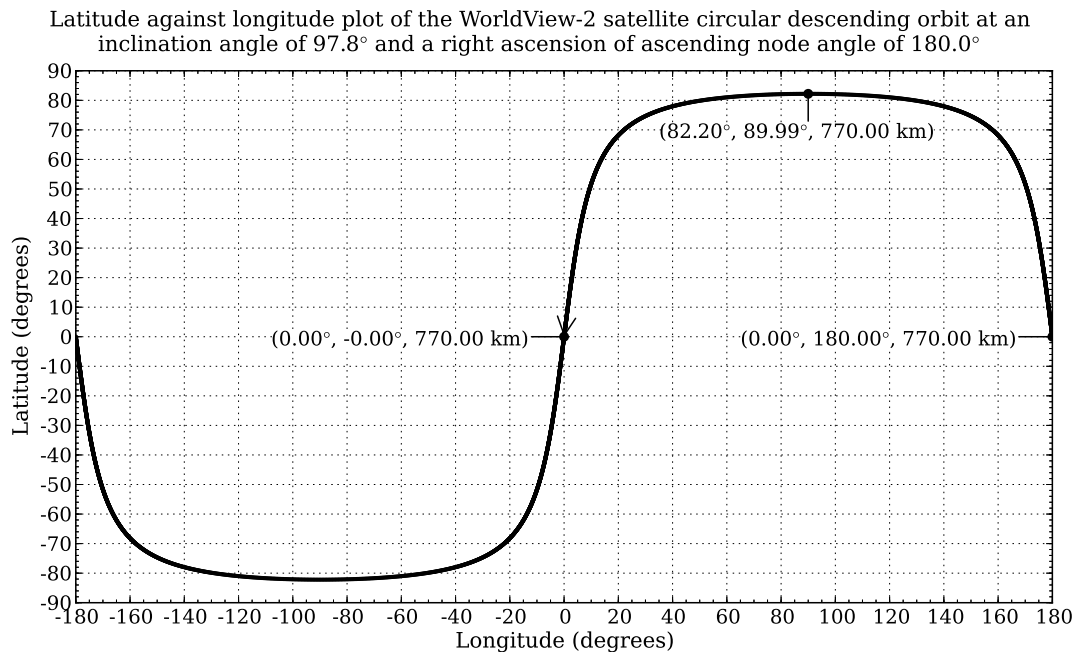


Figure 11.16: The full circular descending retrograde orbit of the WorldView-2 satellite.

The plot in Figure 11.16 indicates that the calculated satellite ground track is symmetrical. The arrow on the plot marks the position of the satellite at the start

of the simulation and the direction in which the satellite is orbiting the earth. The orbit trajectory starts at 0° latitude, 0° longitude and moves downwards along the trajectory path. The longitude location of the ascending node was read off the plot as 180° at the 0° latitude value where the orbit is ascending. The inclination angle for the retrograde orbit was calculated using the 82.2° latitude location of the maximum trajectory. The inclination angle was calculated as being $180.0^\circ - 82.2^\circ = 97.8^\circ$ [16, p. 169]. At each marked point in the trajectory, the orbit altitude was seen to be 770 km.

The results of the calculated satellite attitude, as the satellite orbits the earth, is plotted in Figure 11.17. Figure 11.17 provides plots of the satellite's roll, yaw and pitch angles

Plot of the WorldView-2 satellite attitude roll, pitch and yaw angles against the simulation run time for a circular descending orbit at an inclination angle of 97.8° and a right ascension of ascending node angle of 180.0°

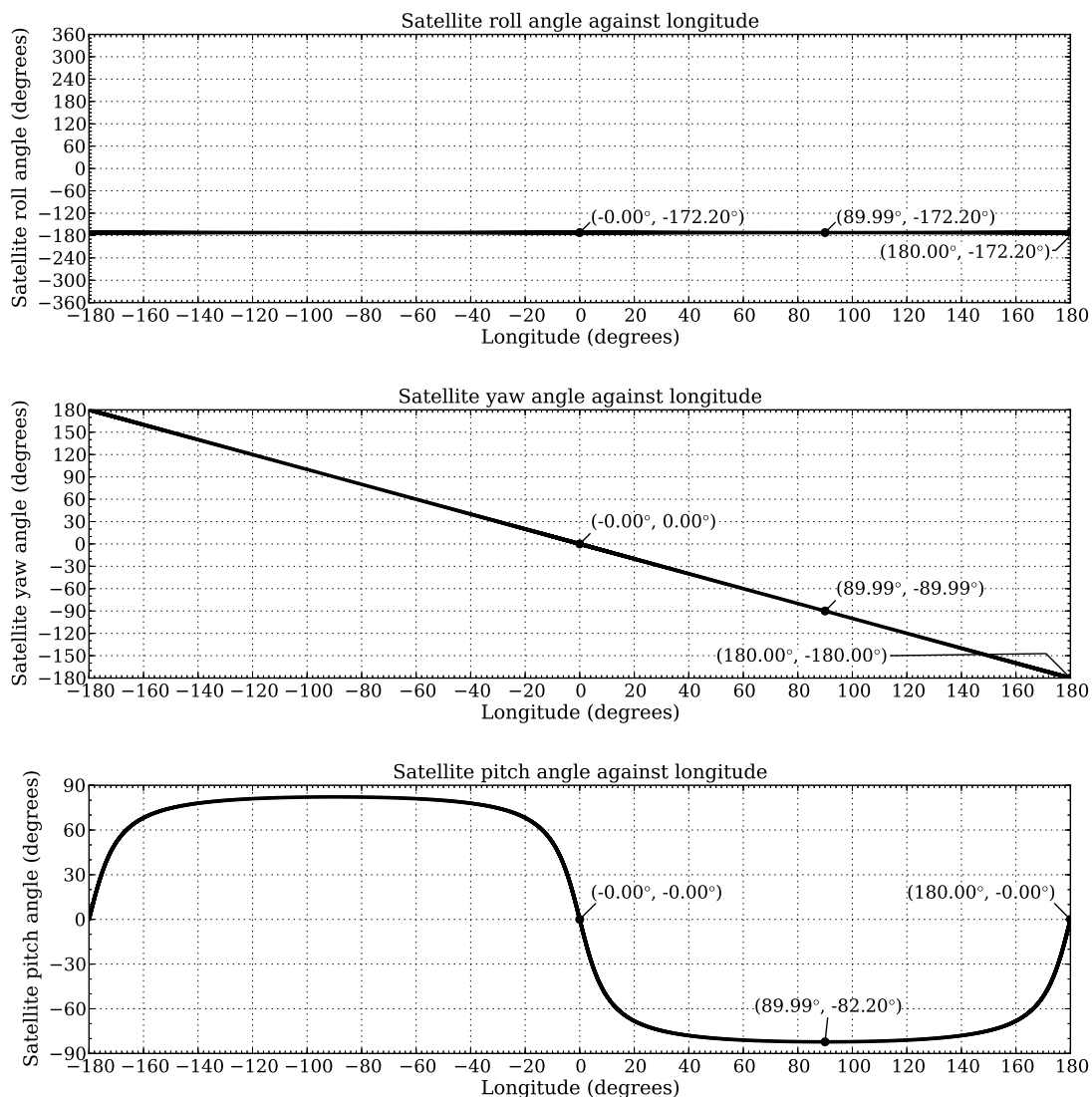


Figure 11.17: WorldView-2 satellite attitude for the calculated full circular descending retrograde orbit.

angles against the longitude angle of the satellite's trajectory. The points corresponding to the ascending and descending orbit nodes and the maximum latitude point have been marked on each of the attitude plots for ease of comparison with Figure 11.16. Each marker provides the longitude and attitude angle at that point in the trajectory.

As seen in Figure 11.17, the satellite attitude yaw and pitch angles are both 0° at the 0° latitude and 0° longitude location on the surface of the earth. The calculated yaw angle is equivalent to the negative of the satellite's longitude location and the calculated pitch angle is equivalent to the negative of the latitude location of the satellite, as shown in Figure 11.16. Figure 11.17 also shows that the satellite roll remains constant for each trajectory point. The orbit inclination angle was set to 97.8° , which means that the satellite rolls by an angle 7.8° off of the 90° starting point. Since the satellite is in a descending orbit, the satellite rolls a further 180° , so as to be pointing in the direction of flight. This results in a roll angle of 187.8° . This roll angle is rewritten as $187.8^\circ - 360.0^\circ = -172.2^\circ$, to fall within the range $[-180^\circ:180^\circ]$.

11.10.3 Analysis of results

When studying the trajectory plot in Figure 11.16, it can be seen that the latitude values decrease when following the trajectory path in the direction pointed by the arrow. This indicates that the orbit has been correctly calculated to descend on the sun side of the earth. As expected, the calculated orbit trajectory was symmetrical about the 0° latitude and longitude lines, which confirmed that the orbit is circular. The longitude value at the ascending node was also observed to correspond with the set RAAN angle. Furthermore, the trajectory calculated results showed that the desired orbit inclination angle of 97.8° had been achieved. It was also observed that the desired orbit altitude of 770 km had been achieved. From the plot in Figure 11.16 it was concluded that the satellite orbit latitude and longitude trajectory path had been calculated correctly.

When studying the yaw angle plot in Figure 11.17, it was seen that, as expected, the satellite yaw angle for a given point along the orbit trajectory was the same as the negative of the satellite trajectory longitude value at that point. Similarly, as expected, the pitch angle for a given point along the orbit trajectory was noted to be the same as the negative of the satellite trajectory latitude value at that point. The roll angle, as expected, remained constant and, as shown by the roll angle calculations, accounted for both the orbit inclination angle and the orbit direction. From the plot in Figure 11.17 it was concluded that the satellite attitude had been calculated correctly.

From the plotted results it was concluded that the OSPA satellite's full orbit trajectory and attitude had been implemented correctly in OSPA and met all the specified parameters. The full circular orbit trajectory algorithm provided a good model of the satellite orbit trajectory and attitude.

11.11 Satellite orbit trajectory between coordinates

11.11.1 Test description

This test point validates the calculation of the satellite trajectory between two points. OSPA provides the functionality to define the satellite forward motion as a direct path between two latitude and longitude coordinates, as discussed in Chapter 8. To evaluate the satellite trajectory between two points, the position and attitude of the satellite, as calculated for the satellite trajectory, are plotted and analysed. The plots were inspected to ensure that the flight direction and the start and stop locations were correct and to ensure that the satellite always pointed to the centre of a perfectly spherical earth target and remained perpendicular to the satellite track. The earth was modelled by a spherical target, Figure 11.1e, which had a radius of 6378.1 km and was assumed to be stationary.

For this test only the satellite trajectory was of interest and not the sensor output image. The orbit algorithm was set to calculate a direct orbit trajectory path between two latitude and longitude locations. For this test the satellite trajectory was set to descend between the locations $(31^{\circ}27'N, 4^{\circ}28'E) = ((31 + (27/60))^{\circ}N, (4 + (28/60))^{\circ}E) = (31.45^{\circ}N, 4.47^{\circ}E)$ and $(31^{\circ}27'S, 4^{\circ}28'W) = ((31 + (27/60))^{\circ}S, (4 + (28/60))^{\circ}W) = (31.45^{\circ}S, 4.47^{\circ}W)$. These location points were chosen such that the trajectory would have the same orbit inclination angle as the WorldView-2 satellite orbit plotted in Figure 11.16. The orbit trajectory was set to an altitude of 770 km over the surface of the earth target. The orbit was calculated relative to a reference point set to be at the top of the earth ($90^{\circ}0'0''N, 0^{\circ}0'0''E$). The satellite attitude was set to be calculated for the yaw-pitch-roll Euler angle rotation order.

The calculated trajectory was expected to be a direct path between two coordinates with a constant altitude above the surface of the earth. As with the full orbit, the satellite yaw attitude angle was expected to be equivalent to the negative of the satellite's longitude position. This yaw angle would allow the satellite to point to the correct longitude location on the surface of the earth. The satellite pitch attitude angle was expected to be equivalent to the negative of the satellite's latitude position. This pitch angle would allow the satellite to look down towards the centre of the earth. The satellite roll attitude angle was expected to correspond to the orbit inclination angle and take into account the orbit direction. This would allow the satellite sensor to be aligned perpendicularly to the satellite track, as well as to face forward along the track. The satellite roll angle should remain constant throughout the trajectory.

11.11.2 Results

The result of the calculated satellite latitude and longitude trajectory path between two coordinates on the surface of the earth is plotted in Figure 11.18. The start and stop

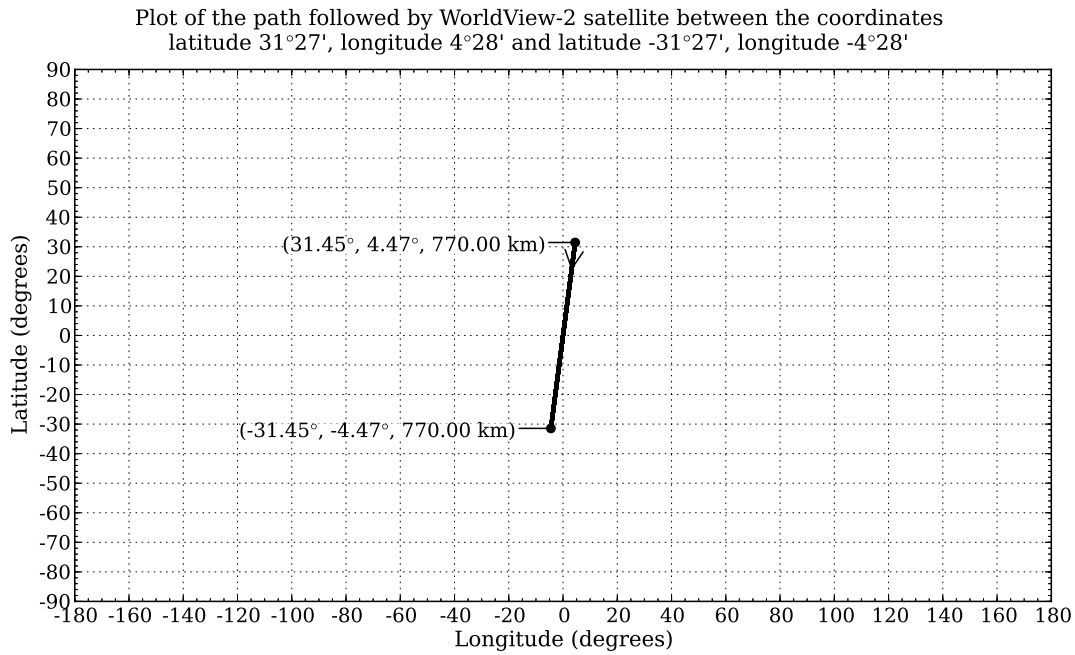


Figure 11.18: WorldView-2 satellite latitude and longitude for the calculated orbit trajectory between two coordinates.

trajectory locations are marked on the plot. Each marker provides the latitude, longitude and orbit altitude value at that point in the trajectory.

Figure 11.18 indicates that the satellite travels in a direct path between the two specified locations. The arrow on the plot marks the position of the satellite at the start of the simulation at 31.45° latitude and 4.47° longitude, and the direction in which the satellite is moving. The satellite trajectory starts above the 0° latitude line and moves downwards to below the 0° latitude line, to the end location at -31.45° latitude and -4.47° longitude. At each marked point in the trajectory the orbit altitude was seen to be 770 km.

The result of the calculated satellite attitude, as it orbited the earth, were plotted in Figure 11.19. Figure 11.19 provides plots of the satellite roll, yaw and pitch angles against the longitude angle of the satellite trajectory. The points corresponding to the beginning and ending points of latitude have been marked on each of the attitude plots, for ease of comparison with Figure 11.18. Each marker provided the longitude and attitude angle at that point in the trajectory.

The yaw and pitch plots in Figure 11.19 show that the calculated yaw angle is equivalent to the negative of the satellite's longitude location and the calculated pitch angle is equivalent to the negative of the latitude location of the satellite. From the roll plot in Figure 11.19 it is seen that the satellite roll remains at a constant angle of -172.2° for each trajectory point. The orbit inclination angle was set 97.8° , which meant that the satellite rolled by an angle 7.8° off from the 90° starting point. Since the satellite

Plot of the WorldView-2 satellite attitude roll, pitch and yaw angles along the path followed by a satellite between the coordinates latitude $31^{\circ}27'$, longitude $4^{\circ}28'$ and latitude $-31^{\circ}27'$, longitude $-4^{\circ}28'$

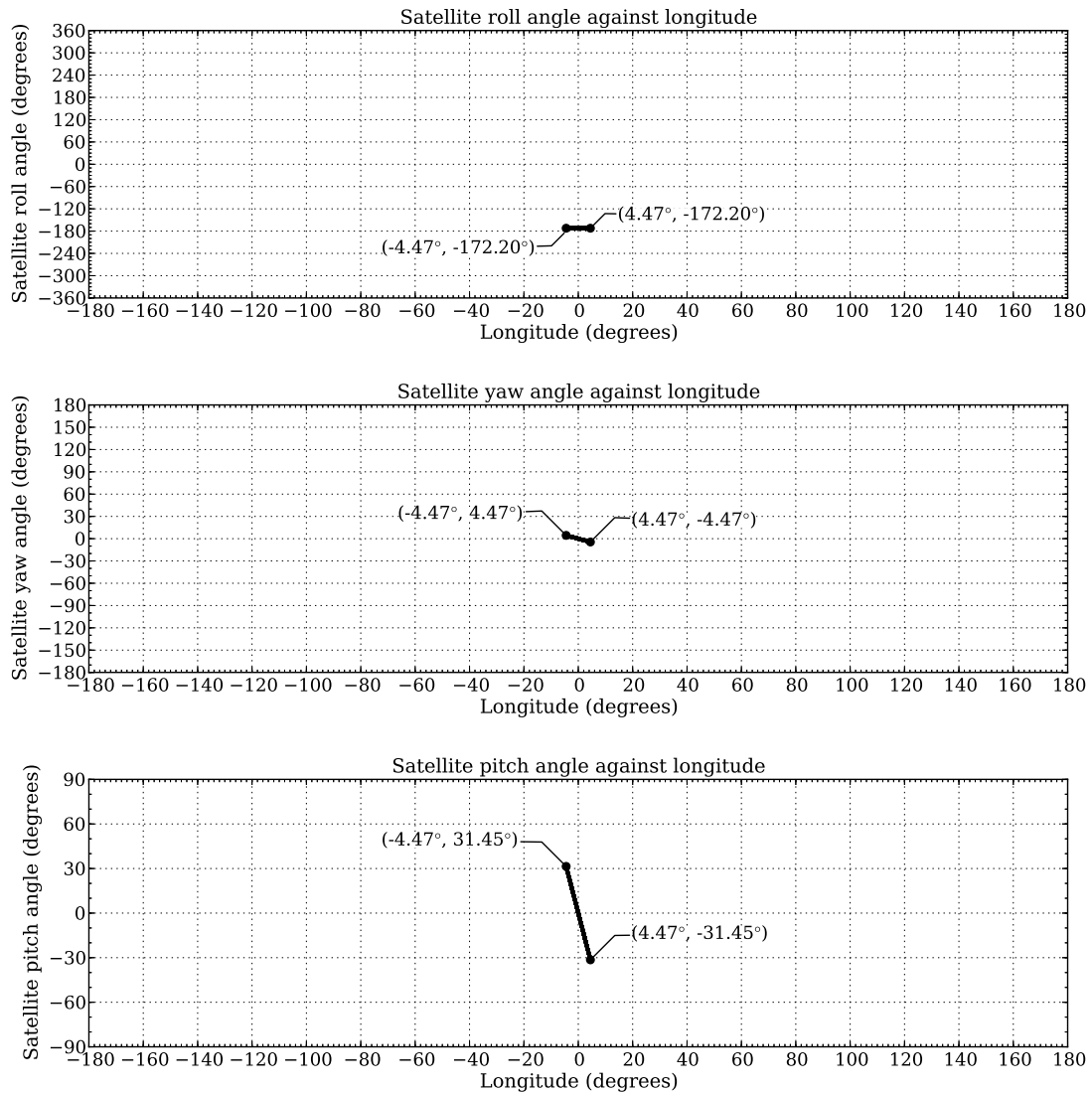


Figure 11.19: WorldView-2 satellite attitude for the calculated orbit trajectory between two coordinates.

was in a descending orbit, the satellite rolled a further 180° , so as to be pointing in the direction of flight. This resulted in a roll angle of 187.8° . This roll angle was rewritten as $187.8^{\circ} - 360.0^{\circ} = -172.2^{\circ}$, to fall within the range $[-180^{\circ}; 180^{\circ}]$.

11.11.3 Analysis of results

When studying the trajectory plot in Figure 11.18 it was seen that the latitude values decreased when following the plot in the direction pointed by the arrow. This indicated that the orbit direction had been calculated correctly. The results also showed that the trajectory path was a direct path between the specified trajectory beginning (31.45°N , 4.47°E) and end (31.45°S , 4.47°W) locations. It was also observed that the

desired orbit altitude of 770 km had been achieved. From the plot in Figure 11.18 it was concluded that the satellite orbit trajectory path had been calculated correctly.

When studying the yaw angle plot in Figure 11.19, it was seen that, as expected, the satellite yaw angle for a given point was the same as the negative of the satellite trajectory longitude value at that point. Similarly, as expected, the pitch angle was noted to be the same as the negative of the satellite trajectory latitude value at that point. The roll angle, as expected, remained constant and accounted for both the orbit inclination angle and the orbit node direction. From the plot in Figure 11.19 it was concluded that the satellite attitude was calculated correctly.

From the plotted results it was concluded that the satellite orbit trajectory and attitude calculation for a trajectory between two coordinates had been correctly implemented in OSPA and that it met all the specified parameters. The orbit trajectory algorithm provided a good model of the satellite orbit trajectory and attitude between two locations on the earth.

11.12 Staggered detector layout image correction

11.12.1 Test description

This test validates the stagger correction image processing algorithm. This algorithm corrects the raw sensor output image for discontinuities due to a staggered detector layout. The algorithm assumes when processing the image that a specific set of conditions, as discussed in section 9.3, have been met. The image is corrected for a specific satellite platform velocity. Any deviations in the satellite velocity will cause the image to be processed incorrectly and discontinuities in the target geometry will still be present in the output image. The correctly processed image was expected to show no geometric distortion in the image that were due to a staggered detector array layout. An incorrectly processed image was expected still to exhibit geometric distortion due to a staggered array.

To test the image processing algorithm the sensor was set to image a narrow line target, Figure 11.1c, using the WorldView-2 satellite platform velocity and the detector array configuration described in test section 11.3. The algorithm performance was tested for the scenarios where the satellite platform velocity requirement was met and where it was not met. To test the algorithm when the velocity requirement is not met, the satellite velocity was decreased, which would result in the image processing algorithm under-correcting the image, by correcting for too few staggered lines. The satellite velocity was also increased, which would result in the image processing algorithm over-correcting the image, by correcting for too many staggered lines.

To generate a correctly processed image, the image processing algorithm was set to correct for a WorldView-2 platform velocity of 7471 m/s. The simulated satellite was set to have the same velocity, by setting the satellite to move forward in the scene by 0.4632 m every dwell time of 62 μ s, as discussed in section 6.3.5. Using (9.3.1) and (9.3.2), in section 9.3, the image processing algorithm set the stagger correction to 64 pixels for the panchromatic band image and 24 pixels for the multispectral band images. This matched the DSA stagger distances specified in section 6.3.4.

To produce an under-corrected image the simulated satellite was set to have a velocity of 6723.87 m/s by setting the satellite to move forward in the scene by 0.41688 m which would result in a 10% overlap in the GSD for the dwell time of 62 μ s. The under-corrected processed image was, therefore, expected to show an undercorrection of $((64 \times 0.4632)/0.41688) - 64 = 7.111 \approx 7$ pixels for the panchromatic band image, which had been set to be corrected for 64 pixels. The multispectral band images were expected to show an undercorrection of $((24 \times 0.4632 \times 4)/(0.41688 \times 4)) - 24 = 2.667 \approx 3$ pixels, since the multispectral band images had been set to be corrected for 24 pixels.

To produce an over-corrected image the simulated satellite had been set to have a velocity of 8218 m/s by setting the satellite to move forward in the scene by 0.50952 m, which would result in a 10% gap in the GSD for the dwell time of 62 μ s. The over-corrected image was expected to show an overcorrection of $64 - ((64 \times 0.4632)/0.50952) = 5.818 \approx 6$ pixels for the panchromatic band image, which was set to be corrected for 64 pixels. The multispectral band images were expected to show an overcorrection of $24 - ((24 \times 0.4632 \times 4)/(0.50952 \times 4)) = 2.181 \approx 2$ pixels since the multispectral band images were set to be corrected for 24 pixels.

All interleaving pixels were filled with a zero greylevel value.

The simulation was run with the atmosphere set to the good visibility Tropical atmospheric model, described in section 5.3.1. No noise or detector non-uniformity was included when running this test.

11.12.2 Results

All the simulated WorldView-2 spectral imaging bands imaged the line target, and the resultant images were corrected for stagger distortion. Table 11.8 lists the image processing results for each of the WorldView-2 imaging bands.

To illustrate how the values in Table 11.8 were calculated, the results and calculations for the coastal blue imaging band are discussed in detail. Figure 11.20 shows the output sensor image for the simulated WorldView-2 coastal blue imaging band, and the image results when the image processing algorithm was applied to the raw image.

The top left image in Figure 11.20 shows the sensor output image that is to be corrected for the stagger distortion. The top right image shows no distortion and this confirms

Table 11.8: Pixel distortion present in images processed for different satellite platform velocities for each of the simulated WorldView-2 imaging spectral bands.

Spectral band	Processed image pixel distortion for satellite velocity 7471 m/s	Processed image pixel distortion for satellite velocity 6723.87 m/s	Processed image pixel distortion for satellite velocity 8218 m/s
Panchromatic	0	7	6
Coastal blue	0	3	2
Blue	0	3	2
Green	0	3	2
Yellow	0	3	2
Red	0	3	2
Red edge	0	3	2
NIR-1	0	3	2
NIR-2	0	3	2

Image correction for distortion due to an along-track staggered detector array configuration for the coastal blue band

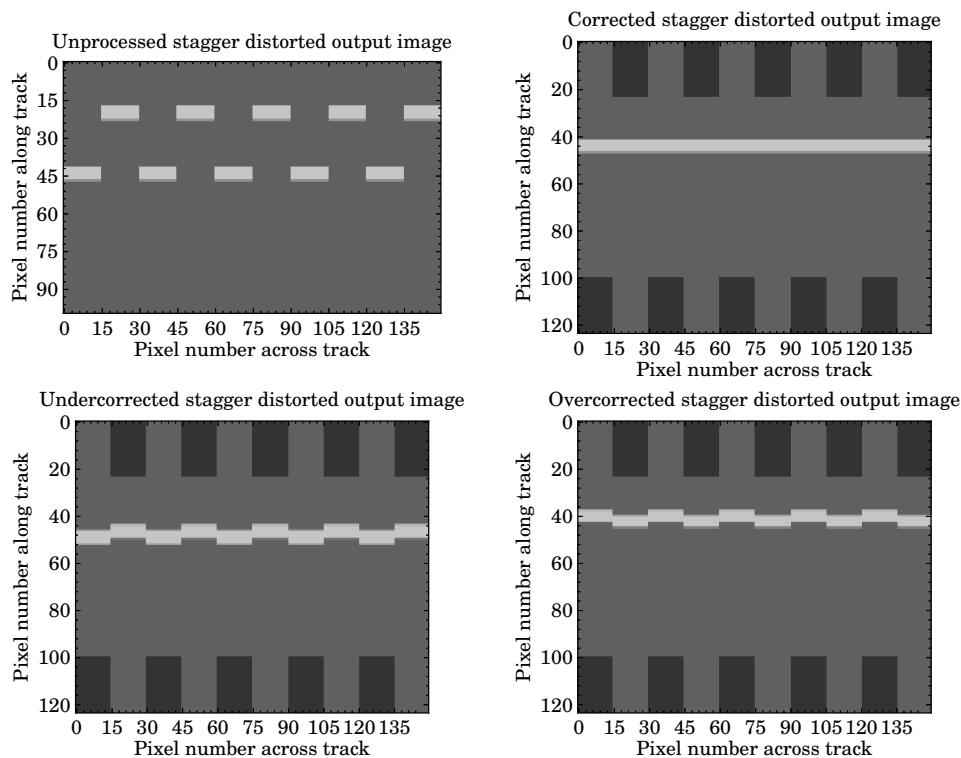


Figure 11.20: WorldView-2 satellite coastal blue band output image and processed images for an along-track staggered detector array layout.

that the image processing algorithm was able to remove the distortion effects of the staggered array correctly. The lower left image illustrates the effects of under-correcting

the image for stagger distortion. The line target still appears to be staggered, since the pixels corresponding to the even numbered DSAs have been shifted along the track by too few image rows. The lower right image illustrates the effect of over-correcting the image for stagger distortion. The line target still appears to be staggered, since the pixels corresponding to the even numbered DSAs have been shifted along the track by too many image rows.

The number of missing or incorrectly added interleaving pixels is determined by examining the DN value plots of the image pixel columns 59 and 60, shown in Figure 11.21. The pixel points in the image that correspond to the same along-track location, and that are to be aligned, are marked. From the plots in Figure 11.21 it can be seen that both the

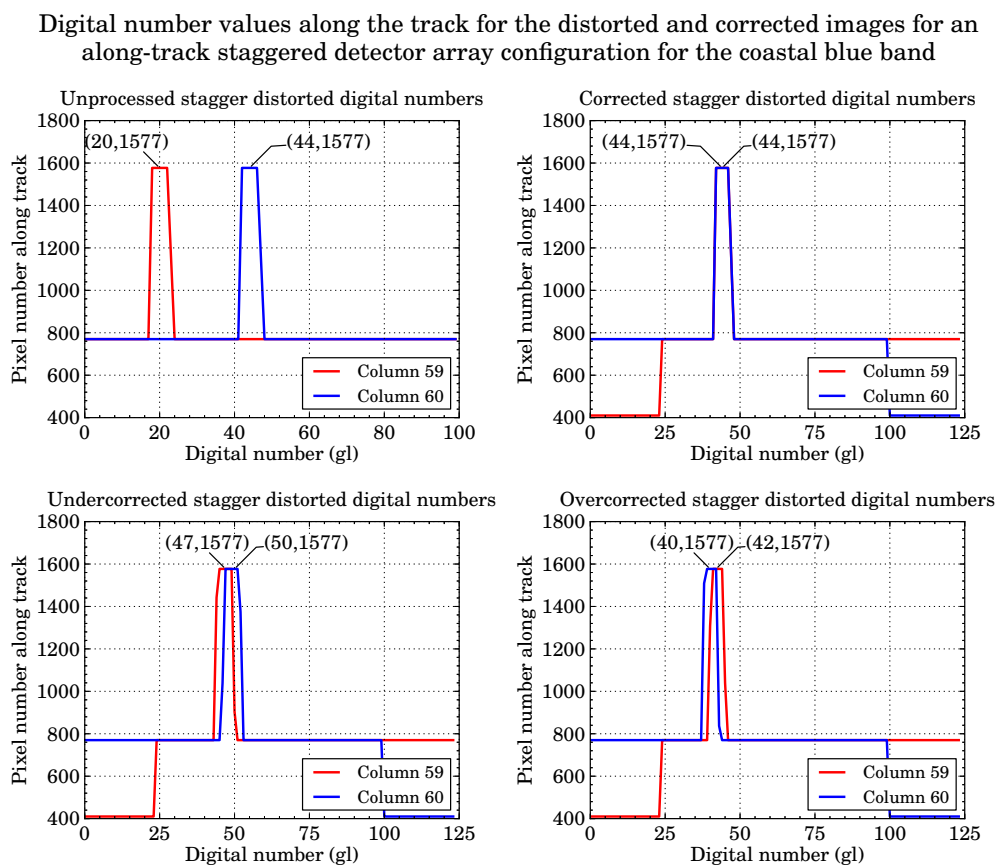


Figure 11.21: Digital number values along a column of image pixels in the output image from a staggered detector array and for a correctly processed, an under-corrected and an over-corrected image.

marked target points align at pixel row 44 in the correctly processed image. This indicates that all stagger distortion has been removed from the image. The under-corrected image shows that the pixels were shifted along the track by $50 - 47 = 3$ pixels too few. The over-corrected image shows that the pixels were shifted along the track by $42 - 40 = 2$ pixels too many.

11.12.3 Analysis of results

As seen in Figures 11.20 and 11.21, the image processing algorithm produced the expected processed output image result. The images appeared corrected or staggered, depending on the satellite's velocity. When comparing the results in Table 11.8 to the expected results, it was seen that the processed images displayed the expected pixel distortion values.

It was concluded that the OSPA image processing algorithm was able to correctly process the sensor's output image for the distortion due to a staggered detector layout. The algorithm also produced the expected uncorrected image distortion that is due to a change in the satellite's velocity.

11.13 Oscillating scan mirror image correction

11.13.1 Test description

This test validates the oscillation correction image processing algorithm. This algorithm corrects the raw sensor output image for the discontinuities caused by an oscillating scan mirror. The algorithm assumes when processing the image that a specific set of conditions, discussed in section 9.4, has been met. The image is corrected by reversing the order of the pixels in the image row for alternating swath scans. A corrected image was expected to show no oscillating scan line distortion.

To test the oscillation correction image processing algorithm, the sensor was set to image a triangular target, similar to Figure 11.1d, where the low reflectance material is on the right half of the triangle. The sensor scanned the triangular target using a descending orbit. This means that the pixels of the low reflectance material (the dark pixels) of the triangular target were expected to appear on the left side of the processed output image, as explained in section 11.9. The detector array configuration and oscillating scan mirror settings described in the across-track oscillating scan mirror test section 11.8 were used for this test.

The swath width was set to be sampled 401 times by the WorldView-2 panchromatic imaging band. According to the different imaging band dwell times, the WorldView-2 multispectral bands sample the swath four times fewer than the panchromatic band does and will, therefore, sample the swath width 100 times. The scan mirror was first tilted so that the sensor started imaging the swath from the left of the swath. The second swath scan line scanned from the right to the left across the track. This swath scan line was the first line in which the target appeared, as illustrated in Figure 11.11, and was the first swath scan line to be corrected. The next across-track scan imaged the target pixels in the correct order, from left to right, which resulted in this scan line not being corrected by the image processing algorithm. The algorithm would be proven to work correctly if

no pixels of the high reflectance material (light pixels) or low reflectance material (dark pixels) in an image row still appeared on the incorrect side of the triangle target image.

The simulation was run with the atmosphere set to the good visibility Tropical atmospheric model, described in section 5.3.1. No noise or detector non-uniformity was included when running this test.

11.13.2 Results

All the WorldView-2 spectral imaging bands imaged the triangular target using the oscillating mirror across-track scanner. The distorted output image and the processed image for the panchromatic imaging band are given in Figure 11.22.

Image correction for distortion of the panchromatic band due to an oscillating scan mirror

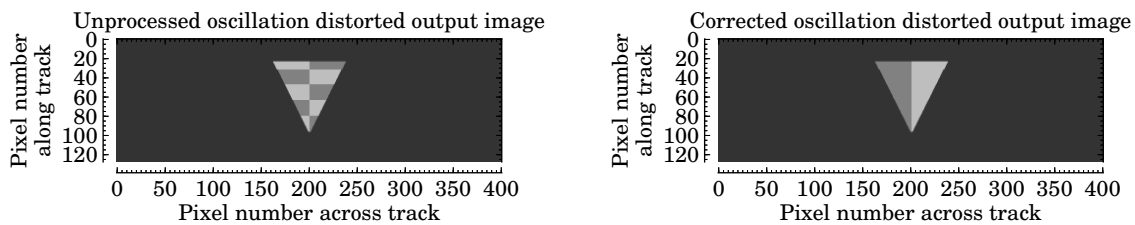


Figure 11.22: Satellite sensor's panchromatic band output image and corrected image for a target imaged using an oscillating scan mirror.

As seen in the unprocessed image of the triangular target, in Figure 11.22, the light and dark pixels of the triangle appear to swap with every swath scan. From visual inspection of the output images, it can be seen that all light image pixels on the left of the distorted image have been processed to appear on the right of the processed image. Similarly, all dark image pixels on the right of the distorted image appear on the left of the processed image.

All multispectral imaging bands displayed the same oscillating scan mirror distortion, and all images were processed correctly to show no distortion.

11.13.3 Analysis of results

The image processing results illustrated that, as expected, all the light image pixels appeared together on the right of the image and the dark pixels on the left. All image results matched the expected results and it can therefore be concluded that the algorithm correctly reverses the alternating image scan lines.

11.14 Conclusion

The OSPA functionality discussed in this document was evaluated to ensure that the system operated correctly. The OSPA functions designed and tested in this study are the in-FOV adjacency effect, the configurable CCD focal plane layout, the $NE\Delta\rho$ for simulating the sensor noise, the ADC, the across-track scanner rotating and oscillating mirrors, the satellite orbit trajectory and the image pre-processing algorithms. A set of test scenarios and targets were designed to test the different functionalities.

The results from the test scenarios showed that the numeric and visual output of the satellite remote sensing system closely matched those of the expected outputs. This indicates that the OSPA functionalities operated correctly within the limitations and operation conditions placed on the functionality. If the operation conditions are not met, it was observed that the system produced the expected error in the output, as shown when testing the image correction algorithm for the stagger detector array.

It was shown that OSPA is able to accurately simulate the required satellite remote sensing system functionality. This study implemented generic functionality, providing the user with the option to provide parameters for the required system. The accuracy of the software model for a specific satellite should be confirmed by comparing the model against real world satellite data.

Chapter 12

Conclusion and Recommendations

In this study a software model was developed that is able to simulate the data acquisition process related to satellite remote sensing for the purpose of image generation. The study focused on the target-atmosphere-sensor relationship in the process. The model was implemented as a space remote sensing application in the OSSIM simulation environment.

To model the satellite remote sensing system, the different system components were identified and analysed. The components considered in this study included the properties of the target, the atmospheric effects, the sensor architecture, scanning methods and the satellite orbit. A generic satellite sensor model was one of the primary focuses of the study and this included a configurable CCD array, a detector noise model and an ADC.

This study also focused on the signature rendering equation used to calculate the at-sensor radiance. The OSSIM signature rendering equation was found to include the necessary terms, with the exception of an adjacency effect term, that were required to model the at-sensor radiance for a satellite sensor. The signature rendering equation was expanded to include a term to describe the in-FOV adjacency effect that is due to aerosol scattering. The study proved that the in-FOV adjacency effect can be modelled as a constant value over the sensor FOV.

The main implementation issues of the model were the lack of available hardware memory and the lengthy computation time (in excess of 20 minutes) required. The main limitation in the current models was the insufficient level of detail with which some features were modelled (a short term practical matter, rather than a fundamental limitation). These limitations can be addressed in future work.

It is recommended that the sensor model be expanded to include a TDI model. The orbit trajectory algorithm should be expanded to calculate orbits other than perfectly circular orbits and the effects of external forces on the satellite orbit trajectory should be considered. It is also recommended that more detailed world scene models be designed.

The software model designed in this study proved to be a good tool with which to simulate a satellite's remote sensing system. The software model provides a framework

which can be further developed to improve on the existing functionality and to include user specific models. Testing of the model showed that all functions operated correctly within the set operating conditions. All objectives of this study were achieved.

List of References

- [1] J. R. Schott, *Remote Sensing: The Image Chain Approach*, 2nd ed. New York, NY: Oxford University Press, Inc., 2007.
- [2] R. A. Schowengerdt, *Remote Sensing: Models and Methods for Image Processing*, 3rd ed. Burlington, MA: Academic Press, 2007.
- [3] T. M. Lillesand, R. W. Kiefer, and J. W. Chipman, *Remote Sensing and Image Interpretation*, 6th ed. Hoboken, NJ: John Wiley & Sons, Inc., 2008.
- [4] *Engineering and Design: Remote Sensing, Engineer Manual EM 1110-2-2907*, Department of the Army, US Army Corps of Engineers, CECW-EE, Washington, DC, Oct. 2003.
- [5] R. D. Fiete, “Image chain analysis for space imaging systems,” *J. Imaging Sci. Technol.*, vol. 51, no. 2, pp. 103 – 109, Mar. 2007.
- [6] C. J. Willers, M. S. Willers, and F. Lapierre, “Signature modelling and radiometric rendering equations in infrared scene simulation systems,” in *Proc. of SPIE Security and Defence 2011*, vol. 8187, Prague, Czech Republic, Sep.19–22, 2011.
- [7] C. J. Willers and M. S. Willers, “OSSIM: Optronics system simulator,” Council for Industrial and Scientific Research, Pretoria, SA, Tech. Rep. CSIR White Paper 6700-PG-103400-01 RPT Rev 5, Apr. 2014.
- [8] *Remote Sensing Tutorial: Fundamentals of Remote Sensing*, Canada Centre for Remote Sensing, 2007. [Online]. Available: http://www.ldeo.columbia.edu/res/fac/rsvlab/fundamentals_e.pdf [Accessed: Mar. 18, 2012].
- [9] C. J. Willers, *Electro-Optical System Analysis and Design: A Radiometry Perspective*, 1st ed. Bellingham, WA: SPIE Press, 2013.
- [10] S. C. Liew. (2001) Principles of remote sensing. Centre for Remote Imaging, Sensing and Processing, National University of Singapore. [Online]. Available: <http://www.crisp.nus.edu.sg/~research/tutorial/rsmain.htm> [Accessed: Mar. 4, 2012].

- [11] N. S. Kopeika, “Optical properties of the atmosphere,” in *A system engineering approach to imaging*, 2nd ed. Bellingham, WA: SPIE Press, 1998, ch. 15.
- [12] S. Y. Kotchenova, E. F. Vermote, R. Levy, and A. Lyapustin, “Radiative transfer codes for atmospheric correction and aerosol retrieval: intercomparison study,” *Appl. Opt.*, vol. 47, no. 13, pp. 2215 – 2226, May 2008.
- [13] L. M. Vhengani, D. Griffith, and M. Lysko, “Campaign for vicarious calibration of SumbandilaSat in Argentina,” in *Proc. of the SAIP 2011*, Saint George Hotel, Irene, Jul.12 – 15 2011, pp. 555 – 560.
- [14] C. Wynn, “An introduction to BRDF-based lighting,” NVIDIA Corporation, Tech. Rep., 2000.
- [15] D. Poli, “Modelling of spaceborne linear array sensors,” Ph.D. dissertation, Swiss Federal Institute of Technology Zurich, Jan. 2005.
- [16] Federal Aviation Administration, “Basic concepts of manned spacecraft design: 4.1.4 describing orbits,” in *Advanced Aerospace Medicine On-line: Section III - Space Operations*, ch. 4, pp. 152–171.
- [17] A. Börner, L. Wiest, P. Keller, R. Reulke, R. Richter, M. Schaepman, and D. Schläpfer, “SENSOR: a tool for the simulation of hyperspectral remote sensing systems,” *ISPRS J. Photogramm. Remote Sens.*, vol. 55, no. 5 – 6, pp. 299 – 312, Mar. 2001.
- [18] E. Vermote, D. Tanré, J. L. Deuzé, M. Herman, J. J. Morcrette, and S. Y. Kotchenova, *Second Simulation of a Satellite Signal in the Solar Spectrum - Vector (6SV)*, 3rd ed., Department of Geography, University of Maryland and Laboratoire d’Optique Atmosphérique and European Centre for Medium Range Weather Forecast, Nov. 2006.
- [19] N. S. Kopeika, “Imaging through the atmosphere for airborne reconnaissance,” *Opt. Eng.*, vol. 26, no. 11, pp. 1146 – 1154, Nov. 1987.
- [20] F. Lei and H. J. Tiziani, “Atmospheric influence on image quality of airborne photographs,” *Opt. Eng.*, vol. 32, no. 9, pp. 2271 – 2280, Sept. 1993.
- [21] T. Updike and C. Comp, “Radiometric use of WorldView-2 imagery,” DigitalGlobe, Technical note Revision 1.0, Nov. 2010.
- [22] M. Lampton and M. Sholl, “Comparison of on-axis three-mirror-anastigmat telescopes,” in *Proc. of the SPIE, UV/Optical/IR Space Telescopes: Innovative Technologies and Concepts III*, vol. 6687, San Diego, CA, Sep.20 2007, pp. 66 870S–1 – 66 870S–8.

- [23] Metallic coatings. brochure. Quantum Coating Incorporated. [Online]. Available: <http://www.quantumcoating.com/pdf/quantum-metallic-brochure.pdf> [Accessed: Nov. 26, 2013].
- [24] R. D. Fiete, *Modeling the Image Chain of Digital Cameras*, ser. Tutorial Texts in Optical Engineering. Bellingham, WA: SPIE Press, 2010, vol. TT92.
- [25] H. Riebeek and R. Simmon. (2009, Sep.) Catalog of earth satellite orbits. NASA Earth Observatory. [Online]. Available: <http://earthobservatory.nasa.gov/Features/OrbitsCatalog/> [Accessed: Mar. 18, 2014].
- [26] J. P. Kerekes and D. A. Landgrebe, “An analytical model of earth-observational remote sensing systems,” *IEEE Trans. Syst., Man, Cybern.*, vol. 21, no. 1, pp. 125 – 133, Jan. – Feb. 1991.
- [27] J. Norgard, “The electromagnetic spectrum,” in *National Association of Broadcasters Engineering Handbook*, 10th ed., E. A. Williams, Ed. Burlington, MA: Focal Press, 2007, pp. 3 – 10, [Online]. Available: <http://dx.doi.org.ez.sun.ac.za/10.1016/B978-0-240-80751-5.50008-4>.
- [28] The layers of earth’s atmosphere. AIRS, NASA Jet Propulsion Laboratory. [Online]. Available: http://airs.jpl.nasa.gov/maps/satellite_feed/atmosphere_layers/ [Accessed: May 20, 2012].
- [29] *US Army space reference text: Chapter 5 Space Environment and Orbital Mechanics*, US Army Space Institute, Fort Leavenworth, KS, Jul. 1993. [Online]. Available: http://www.fas.org/spp/military/docops/army/ref_text/chap5im.htm [Accessed: Jan. 9, 2013].
- [30] R. Carmichael. (2013, Apr.) Public domain aeronautical software. [Online]. Available: <http://www.pdas.com/atmosdownload.html> [Accessed: Oct. 10, 2013].
- [31] (2007, Feb.) The top of the atmosphere. NASA Earth Observatory. [Online]. Available: <http://earthobservatory.nasa.gov/IOTD/view.php?id=7373> [Accessed: Oct. 19, 2012].
- [32] S. S. Fernández de Córdoba. (2004, Jun.) 100km altitude boundary for astronautics. Fédération Aéronautique Internationale Astronautic Records Commission (ICARE). [Online]. Available: <http://www.fai.org/icare-records/100km-altitude-boundary-for-astronautics> [Accessed: Oct. 19, 2012].
- [33] R. N. Clark, G. A. Swayze, R. Wise, K. E. Livo, T. M. Hoefen, R. F. Kokaly, and S. J. Sutley. (2007) USGS digital spectral library splib06a, digital data series 231.

- U.S. Geological Survey. [Online]. Available: <http://speclab.cr.usgs.gov/spectral.lib06/> [Accessed: May 22, 2012].
- [34] “Time delay integration: enabling high sensitivity detection for imaging-in-flow on the imagestream 100 cell analysis system,” Amnis Corporation, Tech. Rep., Feb. 2005.
- [35] H. Stoewer, R. Hartmann, and L.A.J. Baron von Richter, “An advanced methodology for the design process of a satellite.”
- [36] Mechanism of earth observation satellites. ASTER GDS Earth Remote Sensing Data Analysis Center. [Online]. Available: http://gds.aster.ersdac.jspacesystems.or.jp/gds_www2002/seminer_e/e.o.s_e/set_e.o.s_e.html [Accessed: Oct. 17, 2013].
- [37] 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. L. D. B. Paulson, and T. S. Wilkinson, “PICASSO: an end-to-end image simulation tool for space and airborne imaging systems,” *J. Appl. Remote Sens.*, vol. 4, no. 1, Jun. 2010, doi: <http://dx.doi.org/10.1117/1.3457476>.
- [38] “OSSIM software design document,” Council for Industrial and Scientific Research and Denel Dynamics, Pretoria, SA, Tech. Rep. 6700-OSSIM-10000/01 SDD, Nov. 2013.
- [39] C. J. Willers and M. S. Willers, “OSSIM modelling report,” Council for Industrial and Scientific Research and Denel Dynamics, Pretoria, SA, Tech. Rep. 6700-OSSIM-10000/03 RPT, Jan. 2014.
- [40] J. Zhang, X. Zhang, B. Zou, and D. Chen, “On hyperspectral image simulation of a complex woodland area,” *IEEE Trans. Geosci. Remote Sensing*, vol. 48, no. 11, pp. 3889 – 3902, Nov. 2010.
- [41] J. Choi and T. Kim, “Study on spectral transmission characteristics of the reflected and self-emitted radiation through the atmosphere,” in *Multisensor Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on*, Seoul, Korea, Aug.20–22, 2008, pp. 354 – 359.
- [42] C. Cattrall and K. J. Thome, “Exploitation of MODTRAN 4 capabilities to predict at-sensor radiance,” in *Proc. of the SPIE, Optical Spectroscopic Techniques and Instrumentation for Atmospheric and Space Research V.*, vol. 5157, San Diego, CA, Aug.3 2003, pp. 98 – 106.
- [43] A. Berk, G. P. Anderson, P. K. Acharya, J. H. Chetwynd, L. S. Bernstein, E. P. Shettel, M. W. Matthew, and S. M. Adler-Golden, *MODTRAN4 user’s manual*, Air

- Force Research Laboratory and Spectral Science, Inc. and Naval Research Laboratory, Apr. 2000.
- [44] WorldView-2. eoPortal. [Online]. Available: <https://directory.eoportal.org/web/eoportal/satellite-missions/v-w-x-y-z/worldview-2> [Accessed: Jul. 30, 2012].
- [45] E. W. Weisstein. Spherical trigonometry. MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/SphericalTrigonometry.html> [Accessed: Apr. 21, 2014].
- [46] D. Poli, E. Angiuli, and F. Remondino, “Radiometric and geometric analysis of WorldView-2 stereo scenes,” in *Proc. of the ISPRS Commission I Mid-Term Symposium “Image Data Acquisition - Sensors and Platforms”*, vol. XXXVIII-Part 1, Calgary, AB, Canada, Jun.15–18, 2010, pp. 2490 – 2498.
- [47] “Worldview-2,” DigitalGlobe, Data Sheet DS-WV2 06/13, Jun. 2013.
- [48] M. Sholl, M. Kaplan, and M. Lampton, “Three mirror anastigmat survey telescope optimization,” in *Proc. of the SPIE, Space Telescopes and Instrumentation 2008: Optical, Infrared, and Millimeter*, vol. 7010, Marseille, France, Jul.12 2008, pp. 70 103M–1 – 70 103M–11.
- [49] (2013) FSS99. Quantum Coating Incorporated. [Online]. Available: <http://www.quantumcoating.com/fss99.asp> [Accessed: Jul. 23, 2013].
- [50] C. Rivolta, “Airy disk diffraction pattern: comparison of some values of $f/\text{no.}$ and obscuration ratio,” *Appl. Opt.*, vol. 25, no. 14, pp. 2404–2408, Jul. 1986.
- [51] N. Anderson and G. Marchisio, “Worldview-2 and the evolution of the digitalglobe remote sensing satellite constellation introductory paper for the special session on worldview-2,” in *Proc. of SPIE Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVIII*, S. Shen and P. Lewis, Eds., vol. 8390, May8, 2012, pp. 83 900L–1–83 900L–15.
- [52] D. Griffith, private communication, Jul. 2013.
- [53] *IKONOS Instrument/Product description*, GeoEye, 2009.
- [54] *DigitalGlobe WorldView-1 Instrument/Product Descriptions*, DigitalGlobe, Jun. 2009.
- [55] “Fairchild imaging ccd 10121,” Fairchild, Data Sheet CCD10121, Jan. 2013.
- [56] “Toshiba ccd linear image sensor,” TOSHIBA, Data Sheet TCD1001P, Mar. 2002.
- [57] “2098 x 3 tri-linear ccd image sensor,” Eastman Kodak Company, Data Sheet KLI-2113, Jul. 2001.

- [58] (2012, Oct.) Orbital elements. NASA human space flight. [Online]. Available: <http://spaceflight.nasa.gov/realdata/elements> [Accessed: Apr. 10, 2013].
- [59] WorldView-2 satellite sensor. Satellite Imaging Corporation. [Online]. Available: <http://www.satimagingcorp.com/satellite-sensors/worldview-2.html> [Accessed: Oct. 30, 2013].
- [60] G. Cai, B. M. chen, and T. H. Lee, “Coordinate systems and transformations,” in *Unmanned Rotorcraft Systems*, ser. Advances in industrial Control. Springer London, 2011, ch. 2, pp. 23–34.
- [61] P. Radhadevi, S. Solanki, R. Ramchandran, and R. Krishnan, “Pre-processing considerations of IRS-P6 LISS-4 imagery,” *Int. J. Appl. Earth Obs. Geoinf.*, vol. 10, no. 2, pp. 133 – 139, Jun. 2008.
- [62] C. J. Willers, “OSSIM testing procedure,” Council for Industrial and Scientific Research, Pretoria, SA, Tech. Rep. 6700-OSSIM-10000/01 SWTD, Jan. 2014.
- [63] A. Ghosh and P. K. Joshi, “Remote sensing of moving objects,” *Curr. Sci.*, vol. 104, no. 12, pp. 1613–1615, Jun. 2013.
- [64] J. Maurer. (2013, Oct.) Chapter 24: Boost.random-reference. Boost c++ libraries. [Online]. Available: www.boost.org/doc/libs/1_49_0/doc/html/boost_random/reference.html [Accessed: Sep. 16, 2013].
- [65] P. B. Willis and C. H. Hsieh, “Space applications of polymeric materials,” Jet Propulsion Laboratory, California Institute of Technology, Tech. Rep., Oct. 1999.
- [66] A. Anvari, F. Farhani, and K. Niaki, “Comparative study in space qualified paints used for thermal control of a small satellite,” *Iranian J. Chem. Eng.*, vol. 6, no. 2, pp. 50 – 62, (Spring) 2009.
- [67] Az-93 white thermal control, electrically conductive paint/coating. AZ Technology. [Online]. Available: www.aztechnology.com/materials-coatings-AZ-93.html [Accessed: Feb. 16, 2013].
- [68] R. Lyle, P. Stabekis, and R. Stoud, “Spacecraft thermal control,” National Aeronautics and Space Administration, NASA space vehicle design criteria (Environment) NASA SP-8105, May 1973.
- [69] D. Friedman, “Progress and challenges for next-generation high-efficiency multijunction solar cells,” *Curr. Opin. Solid State Mater. Sci.*, vol. 14, no. 6, pp. 131 – 138, Dec. 2010.

- [70] Gallium arsenide, gaas. Almaz Optics, Inc. [Online]. Available: <http://www.almazoptics.com/GaAs.htm> [Accessed: Sep. 6, 2013].
- [71] E. Honkavaara, J. Peltoniemi, E. Ahokas, R. Kuittinen, J. Hyyppä, J. Jaakkola, H. Kaartinen, L. Markelin, K. Nurminen, and J. Suomalainen, "A permanent test field for digital photogrammetric systems," *J. Photogramm. Remote Sens.*, vol. 74, no. 1, pp. 95 – 106, Jan. 2008.
- [72] M. Pagnutti, K. Holekamp, R. Ryan, S. Blonski, R. Sellers, B. Davis, and V. Zanoni, "Measurement sets and sites commonly used for characterization," Lockheed Martin Space Operations, Tech. Rep. NASA/SE-2002-10-00070-SSC, Oct. 2002.
- [73] M. Pagnutti, R. E. Ryan, K. Holekamp, D. O'Neal, K. Knowlton, K. Ross, and S. Blonski, "Stennis space center verification and validation capabilities," Poster presented at CALCON Technical Conference on Characterization and Radiometric Calibration for Remote Sensing, Aug. 2005.
- [74] "Satellite sensor radiometric calibration site," *CSIR Science Scope*, vol. 6, no. 4, p. 46, Nov. 2013.
- [75] S. Blonski, K. Ross, M. Pagnutti, and T. Stanley. (2006, March.) Spatial resolution characterization for aerial digital imagery. Presentation slides. Science Systems and Applications, Inc and NASA Stennis Space Center. [Online]. Available: http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20060019120_2006011205.pdf [Accessed: Oct. 20, 2013].
- [76] M. Pagnutti. (2002, March.) NASA IKONOS radiometric characterization. Presentation slides at JACIE workshop. Lockheed Martin Space Operations, NASA Stennis Space Center. [Online]. Available: http://calval.cr.usgs.gov/JACIE_files/JACIE02/Files/27Pagnut.pdf [Accessed: Jul. 17, 2013].
- [77] (2010, Jan. - 2011, Jun.) Stennis space center. Bing Maps. [Online]. Available: <http://www.bing.com/maps/default.aspx?cp=30.385439223224466~-89.62846845784304&lvl=19&style=a> [Accessed: Feb. 26, 2014].
- [78] (2013, Apr.) 25°29'45.5"S 28°22'25.8"E. Google Maps. [Online]. Available: <https://www.google.com/maps/@-25.4959685,28.3738253,369m/data=!3m1!1e3> [Accessed: Feb. 26, 2014].
- [79] E. W. Weisstein. Gaussian integral. MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/GaussianIntegral.html> [Accessed: Jan. 22, 2014].

- [80] E. W. Weisstein. Gaussian function. MathWorld—A Wolfram Web Resource. [Online]. Available: <http://mathworld.wolfram.com/GaussianFunction.html> [Accessed: Jan. 22, 2014].

Appendices

Appendix A

The Optronic Scene System Simulation

A.1 Description of OSSIM

The development of sophisticated electro-optical equipment, such as infrared missile seekers, thermal imagers and infrared countermeasure systems, requires radiometrically calibrated infrared imaging scene simulators in order to support the evaluation and optimisation of the system performance, for different environmental conditions. OSSIM is an engineering development tool that has been developed to meet this need [7, p. 13]. OSSIM allows the developer to focus on the design of optronics systems, rather than the physics of radiometry. OSSIM is currently used primarily in the defence domain, but can be applied to the civilian world. OSSIM is used by the South African Council for Scientific and Industrial Research (CSIR) and the South African missile company Denel Dynamics. OSSIM is a second generation simulation tool based on Simulator for Imaging Systems (SIMIS), which has a 20 year development legacy [7, p. 114].

OSSIM creates a virtual 3D world which contains elements of the real world scenario being simulated. This complex world scene can include static and moving objects, background scenarios and atmospheric conditions [7, p. 13]. OSSIM supports scalability by allowing any number of static scene objects, moving objects and observer objects (objects imaging the scene) in a single simulation run. All objects are visible to all observer objects [7, p. 114]. The simulation environment is able to accurately simulate the radiometry of a scene, the atmosphere, object kinematics and signal processing at a detailed level [7, p. 13]. OSSIM achieves physics-true radiometric modelling by including both reflected sunlight and thermal self-emittance signature components when rendering an object. This ensures radiometric accuracy over a full spectral band ranging from $0.4\mu\text{m}$ to $14\mu\text{m}$. The atmosphere is modelled accurately with the aid of MODTRAN. Sophisticated object movement kinematics and aerodynamics are implemented in a discrete

time linear, time-varying kinematics and control system library. The OSSIM structure also allows an end-user to integrate user specific signal and image processing algorithms into the simulation environment [7, p. 13].

The OSSIM simulator is written in C++ and is able to run on both Windows and Linux systems [7, p. 113]. OSSIM is a file-driven, command line application that reads XML input files and writes various output files. This approach allows batch processing and automated scenario analysis to take place [7, p. 114]. The user is able to visualise the simulation outputs using tools outside the core OSSIM application, such as Python, GnuPlot, Matlab or any other user selected tool, to plot the output file results [7, pp. 13 and 114]. The OSSIM code structure employs a core library with interfaces so that a developer is able to implement specialist models for their specific application [7, p. 113].

A.2 OSSIM code structure

The OSSIM code structure is divided into the Blue Code, Red Code and Green Code as illustrated in Figure A.1.

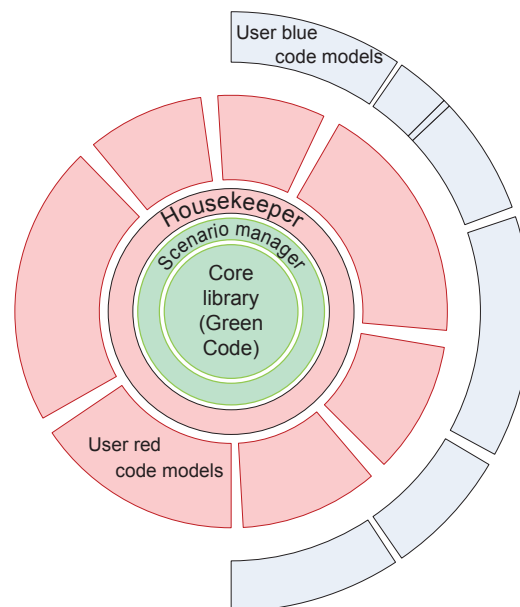


Figure A.1: OSSIM Structure with Green Code at the Core, Red Code Models and User Blue Code Specialist Models (image sourced from [7, p. 113]).

The Green Code is the OSSIM core library module. This code includes general purpose infrastructure elements such as physics, radiometry, the scene graph, the atmosphere, object base classes, rendering and rasterisation and time synchronising. The Green Code provides the functionality to integrate the Red Code and Blue Code modules to work

together. The Green Code functionality is not linked to user specific requirements and can therefore be used by any user [7, pp. 113-114].

The Red Code and Blue Code are the application codes which cover the specialist application and product domain models required by a user or client. Application specific models can include static objects, such as specific trees, moving objects such as aircraft and observer objects such as sensors on a missile or satellite [7, p. 114]. OSSIM provides fully working, high-performance, generic Red Code example models in order to demonstrate how to develop new Red or Blue Code models in the simulation environment [7, p. 114].

A.3 Image rendering in OSSIM using the Green, Red and Blue Code

The Green Code is used to render the image according to generic rules to yield an image unaffected by sensor artefacts. The Red Code and Blue Code are then used to perform the sensor-specific image artefact rendering [7, p. 56]. Figure A.2 shows the roles of the Green Code, Red Code and Blue Code in image rendering.

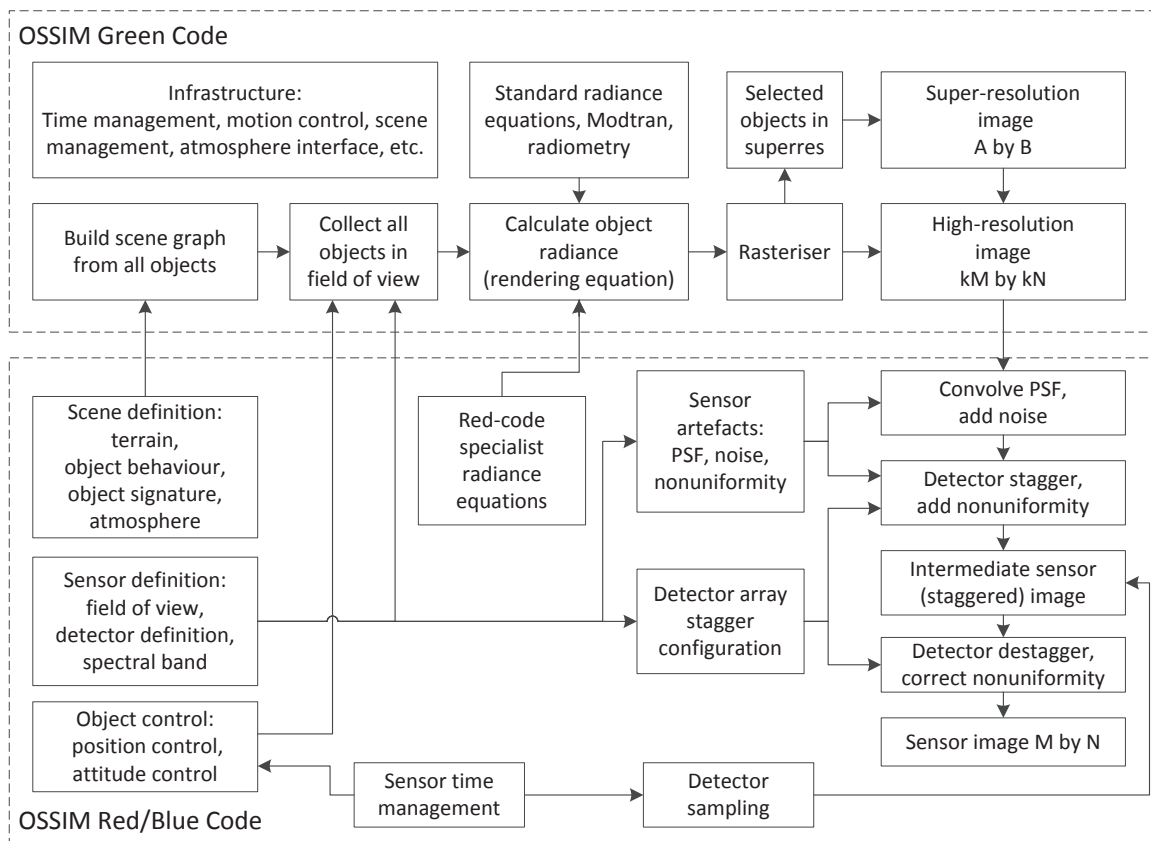


Figure A.2: Roles of Green Code, Red Code and Blue Code in image rendering in the OSSIM environment (image sourced from [7, p. 56]).

The flow diagram in Figure A.2 illustrates the image rendering framework for a linear vector detector used in a linear scan infrared camera or a satellite sensor [7, p. 56].

A.4 High resolution and sensor images

To render a scene in the OSSIM environment, the 3D targets in the object space are mapped into 2D shapes in the image plane. During this transformation process the 3D spatial resolution of the surface of the object is transformed into a 2D irradiance pattern in the focal plane [7, p. 23]. The continuous smooth edged targets are then represented by discrete image pixels. This spatial discretisation results in aliasing, and in some cases radiometrically inaccurate representations of objects [7, p. 54]. The scene image is, therefore, initially rendered in a high spatial resolution image in order to render a radiometrically accurate representation of the target and to perform jagged edge anti-aliasing [7, p. 54]. The high resolution image, generated by the Green Code, is down-convolved with the sensor PSF, using Red Code or Blue Code, to produce a lower resolution sensor image of the appropriate size [7, pp. 54 and 57].

Figure A.3 illustrates the high resolution and sensor images used in the OSSIM and OSPA image generation process.

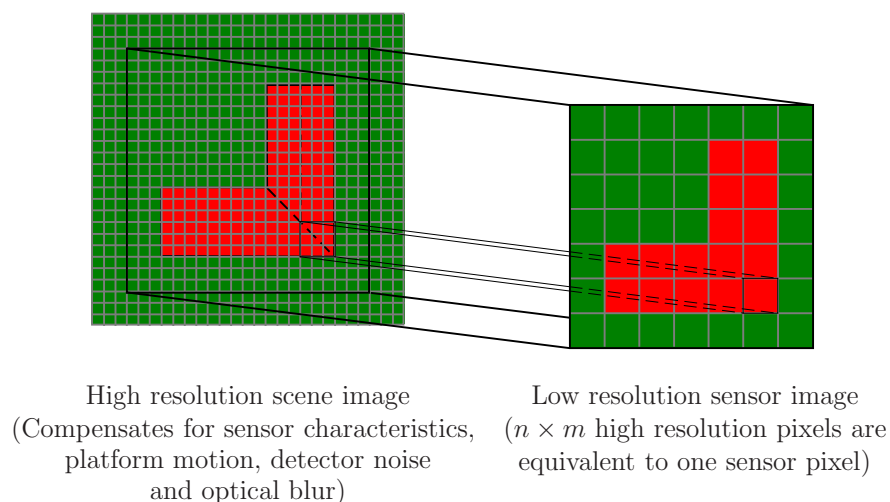


Figure A.3: High resolution and sensor images used in the image rendering process. The high resolution image includes the scene content and atmospheric effects. The high resolution image is resampled to a lower resolution to form the sensor image. The sensor image includes the sensor artefacts and degradation effects.

The high resolution image is considered to be an ideal image of the scene visible in the FOV of the sensor [7, p. 68]. This high resolution, high quality image includes the scene content and atmospheric effects and allows the user application sensor modelling code to model effects such as noise, optical degradation, scanning effects and electronic statistical

variations [7, pp. 23 and 54]. The sensor image includes sensor artefacts or degradation effects [7, p. 54].

The high resolution image is rendered several times, typically three to five times, larger than the sensor image resolution in order to support down-convolution with the sensor PSF and in so doing minimise the aliasing effects in the sensor pixel image [7, pp. 54 and 56-57]. Each sensor pixel consists of an odd number of high resolution image pixels, in order to ensure that there is a centre pixel when convolving the sensor PSF.

A.5 Random number generators used in OSSIM

OSSIM makes use of two random number generators to generate sets of detector noise and non-uniformity values in the sensor model. The two generators used are the subtract-with-carry-01 and Mersenne Twister pseudo-random number generator engines, from the C++ Boost library [64]. The generated random numbers are set to have a Gaussian distribution.

A.6 OSSIM coordinate systems

OSSIM object movement and image rendering takes place in a Local-Level coordinate system with arbitrary reference coordinates. The Local-Level coordinate system can be either an East-South-Down (ESD) or North-East-Down (NED) coordinate system. For the ESD coordinate system, positive x points East, y points South and z points down. For the NED coordinate system, positive x points North, y points East and z points down [39, p. 3.2.1].

Objects in the world have their own body coordinate systems, often with gimbal coordinate systems relative to the body. The object body coordinate system is manipulated relative to the world to orientate the object. Similarly, the object must manipulate the internal gimbal coordinate systems to achieve the specified objectives.

The satellite orientation in OSPA is the same as the observer objects in OSSIM, and is illustrated in Figure A.4 [39, p. 3.2.3].

Following the OSSIM convention, the sensor focal plane is placed in the y-z plane of the coordinate system and positive z points downwards towards the earth. The object body coordinates must be adjusted to pitch the satellite down towards the earth, to roll the satellite to point in the direction of the along-track motion and to yaw the satellite to view the scene off nadir. The gimbal coordinates must be adjusted to alter the satellite sensor's line-of-sight.

All coordinate transformation operations are performed internally to the object model, after which the object is presented to the world.

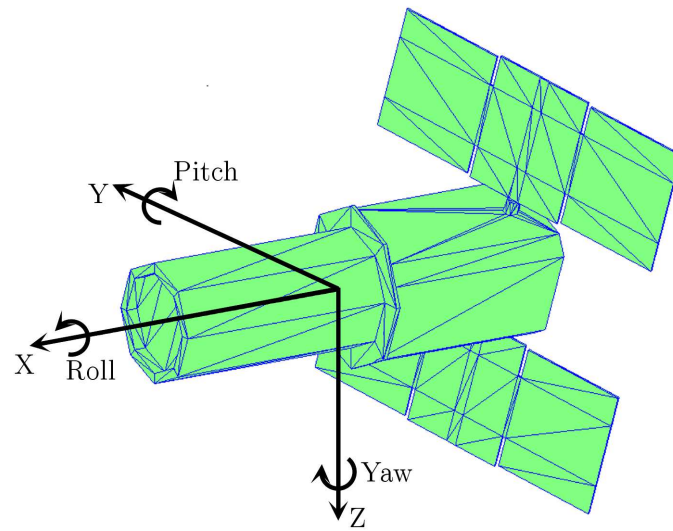


Figure A.4: Satellite orientation in the simulation environment.

Appendix B

The Satellite Body

B.1 Description of a satellite body model

All the objects used in the simulation are visible to other objects and, hence, are required to have a 3D mesh body model. For this study, a simple generic satellite body had to be designed. The body model had to include a basic representation of a satellite bus, payload and solar panels. The radiometric properties of the body model should also represent some of the key materials used in the construction of a spacecraft.

For this study, the generic earth observation satellite body was represented by a simplified model of the WorldView-2 satellite body. The WorldView-2 satellite body is 4.3 m tall, 2.5 m across and 7.1 m across the deployed solar array [44], [47], [59]. Figure B.1 illustrates the primary components of the WorldView-2 satellite architecture.

A simplified geometric model, that is easily recognisable as an earth observation satellite, was designed. The geometry of the generic satellite body model consists of two flat rectangular solar panels, a box shaped bus and a cylindrical shaped payload.

Common materials used in the construction of a satellite body were used to define the radiometric properties of the body model. A satellite is constructed from a number of different materials which include aluminium, titanium, steel and polymeric materials, such as thermal control paints. Composite materials of graphite fibres and polymer matrix resins have become standard materials used in the construction of spacecraft [65].

When assigning the radiometric properties to the satellite body, it was assumed that only two materials are visible on the body. The first of these materials is white thermal control paint, coating the satellite body. The second material is the solar cells, making up the satellite's solar panels. The assumption that only two materials are visible on the satellite body is sufficient for this study, since only a basic satellite body model is required.

The body model is not used as a target in any of the simulations and, therefore, does not need to be modelled in greater geometric or radiometric detail.

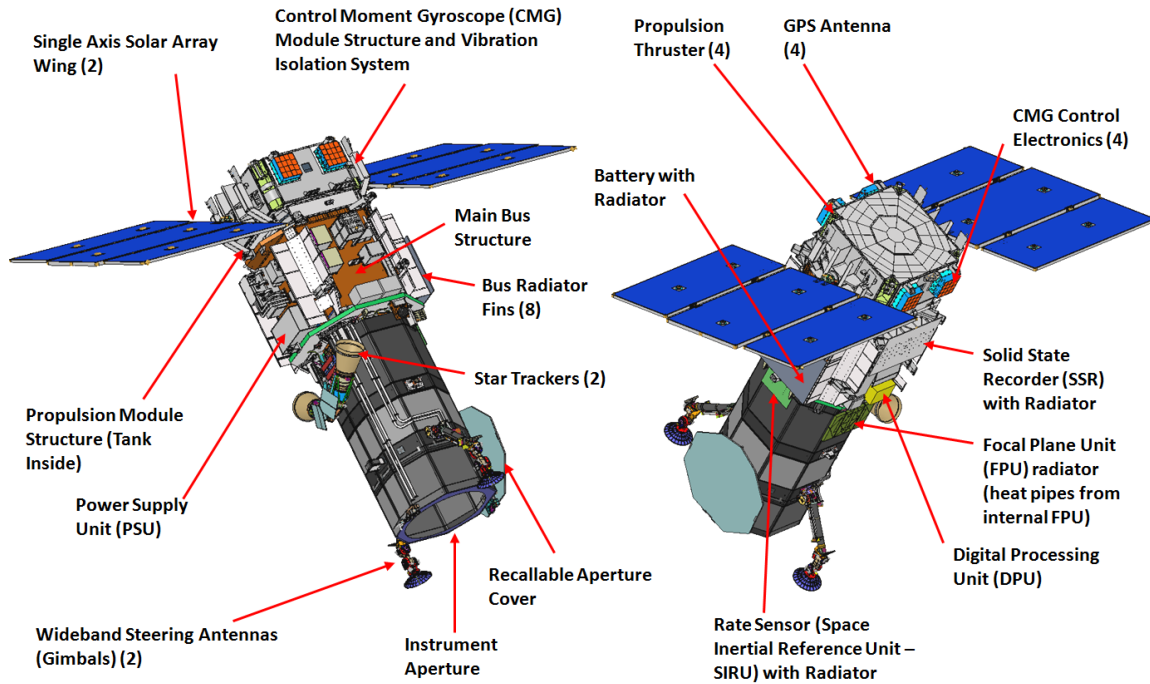


Figure B.1: Primary components of the WorldView-2 satellite architecture (Image sourced from [51]).

B.2 The satellite body model in the simulation environment

An object model in the simulation environment consists of a number of flat polygons forming a 3D mesh model of the object. Each polygon in the model is assigned a set of material properties. The polygon material properties are set for each of the wavelength regions at which a sensor will view the object. The polygon properties include, among others, the material type, emissivity, transmittance, reflectance and thermal properties [7, p. 67].

The satellite body 3D mesh model was designed using Blender and Poledit. Blender is used to construct the 3D mesh model. Poledit is used to assign radiometric properties to each of the polygons in the 3D mesh model. The Poledit software was designed specifically for creating object models that are used in the OSSIM simulation environment [7, p. 67]. Figure B.2 shows the satellite body model as it appears in the simulation environment.

The radiometric properties of typical white thermal control, electrically conductive paint are assigned to the body polygons [66], [67], [68, pp. 6-8]. The radiometric properties of a multi-junction solar cell are assigned to the satellite solar panel polygons [69], [70]. Due to these radiometric properties of the polygons, the solar panel polygons appear darker than the body polygons in Figure B.2. Polygons in shadow also appear darker in the image.

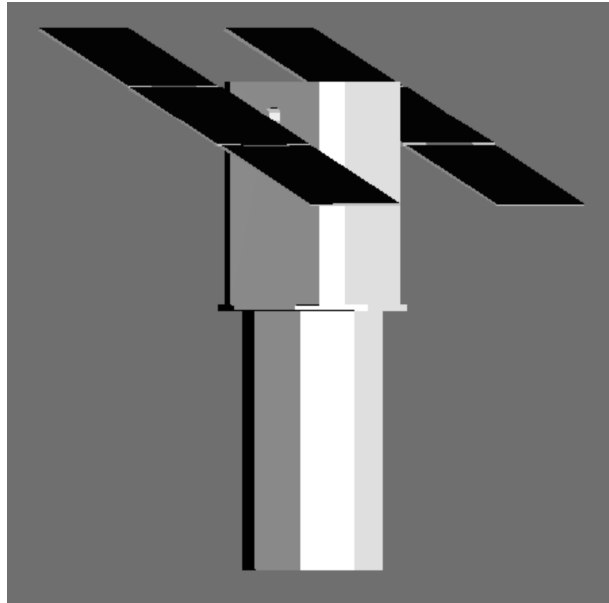


Figure B.2: Satellite body model as seen in the simulation environment, in the visible band.

Appendix C

Test Targets

C.1 Test targets used to validate satellite image data

A set of test targets was designed in order to evaluate the different functionalities in the satellite sensor model. In the OSPA simulation environment, the test target can be a standard calibration target or any target of the user's own making. The test targets are modelled as a set of polygons. Each polygon is assigned a set of radiometric properties and characteristics that match those of a real world material. The targets are designed in the same manner as the mesh model of the satellite body, as discussed in Appendix B. Examples of how these targets are used is shown in Chapter 11.

Examples of real world satellite image test targets are found at the Sjökkulla test field, the Stennis Space Center verification and validation site and at the calibration and validation site at Paardefontein [71], [72], [73], [74].

C.1.1 Targets at the Sjökkulla test field

The Sjökkulla test field is located in a rural area at Sjökkulla ($60^{\circ}14'31.4''\text{N}$, $24^{\circ}23'0.4''\text{E}$) in Kirkkonummi. This test field consists of an image-quality-test field and a geometric-calibration field. The image-quality-test field consists of both permanent and transportable test targets used for radiometric and spatial calibration and testing [71]. In this study, only two of the transportable targets are of interest, namely the greyscale and Siemens star targets. The Siemens star is also known as a radial edge target [73]. Images of these test targets are shown in Figure C.1.

The greyscale target in Figure C.1a consists of eight reflectance reference target blocks. Each block is made from matte tarpaulin with a size of $5\text{ m} \times 5\text{ m}$. The greyscale includes a 5%, 10%, 20%, 25%, 30%, 45%, 50% and 70% reflective reference target block. Sensors with a GSD of 0.5 m or less are able to use this greyscale for calibration [71].



(a) Portable greyscale target.



(b) Portable Siemens star target.

Figure C.1: Transportable test targets found at the Sjökkulla test site (Images from [71]).

The Siemens star in Figure C.1b consists of 19 sectors. Each sector has an arc angle of 10° . Black and grey poly-acrylic fabric is used to make the sectors. The Siemens star has a radius of 6.8 m. Sensors with a GSD of 0.25 m or less can use this target to evaluate the image MTF. The Sjökkulla Siemens star has a constant reflectance in the visible and NIR spectral ranges [71].

C.1.2 Targets at the Stennis Space Center verification and validation site

The Stennis Space center verification and validation site in Mississippi ($30^\circ 23' 7.58''\text{N}$, $89^\circ 37' 42.49''\text{W}$) provides a number of natural and man-made targets. The targets can be used for geometric, spatial and radiometric characterisation [72]. Figure C.2 illustrates three of the test targets found at the Stennis Space center verification and validation site.



(a) Radial edge siemens star target.



(b) Tarpaulin edge target.



(c) Fixed-edge target.

Figure C.2: Test targets found at the Stennis Space Center (Images sourced from [75], [76] and [77]).

Figure C.2a is an image of the Siemens star found at the site. The Siemens star has a

radius of 130 m and has an arc angle greater than 90° . Each of the sectors in the star has a width of approximately 4 m that tapers down to less than 10 cm. The target has been painted on concrete [73]. This target is used to evaluate the system MTF.

Figure C.2b shows the tarpaulin edge target. Two tarpaulin blocks of $20\text{ m} \times 20\text{ m}$ are placed next to each other to form an edge. This configuration allows the target to be used for both spatial and radiometric characterisation. The radiometric tarpaulin target has a reflectivity of 3.5% and 52%. Sensors with a GSD of 4 m or less are able to use this target [72].

The fixed-edge targets, shown in figure C.2c, are edge patterns painted on a concrete slab [72]. Dark paint of 5% reflectivity and light paint of 50% reflectivity, in the visible spectrum, were used to paint these targets [72]. The edge targets consist of two black and white edge patterns, where one pattern lies in a north-south orientation and the other in an east-west orientation [72], [73]. Each edge pattern consists of a $10\text{ m} \times 20\text{ m}$ black and white rectangle. The target pattern has a total dimension of $20\text{ m} \times 40\text{ m}$ [72], [73]. The target pattern is painted twice, where the top target is rotated by 3.7° to account for different imagery coordinate systems [72]. Sensors with a GSD of 1 m or less are able to use these targets [72], [73]. These targets are used for MTF measurements.

C.1.3 Targets at the Paardefontein calibration site

A man-made calibration site was built at Paardefontein, 50 km north of Pretoria ($25^\circ 29' 45.5''\text{S}$, $28^\circ 22' 25.8''\text{E}$). The calibration site has a $10\,000\text{ m}^2$ paved reference target. The bricks used to pave the target area are uniform in both colour and texture. This target is used for vicarious radiometric calibration of satellite and airborne sensors [74]. Figure C.3 shows the Google map image of the calibration target at Paardefontein.



Figure C.3: Paardefontein calibration and validation test target (Image from Google Maps [78]).

C.2 Test target models used in the simulation environment

A set of test targets was designed for geometric and spatial evaluation and sensor radiometric characterisation. The set of targets designed includes models of real world targets and targets designed for specific tests.

Note that all target images shown in this appendix have been scaled to have a maximum grey level of 255 and do not include any sensor or atmospheric effects.

C.2.1 Greyscale and edge test targets

A combined greyscale and edge target set is designed to be used for both spatial and radiometric characterisation. This target consists of a greyscale, a fixed edge target and a tarpaulin edge target. Figure C.4 shows the greyscale and edge test targets as they appear in the simulation environment.

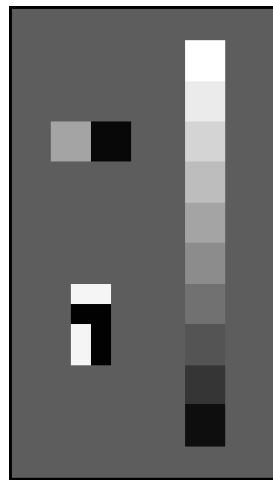


Figure C.4: The tarpaulin edge target (top left), painted edge target (bottom left) and a greyscale bar target (right). The targets are placed on concrete. These targets can be used to validate the radiometric output and spacial frequency of the sensor.

The greyscale target is the target on the right, in Figure C.4. The greyscale target consists of ten blocks of reflective ‘tarpaulin’. The reflectivity of the tarpaulin blocks ranges from 5% at the bottom of the target, to 95% at the top of the target. The reflectivity increases by increments of 10%. Each block is 20 m×20 m in size.

The tarpaulin edge target is the target on the top left, in Figure C.4. The tarpaulin edge target consists of a 55% and a 3.5% reflective block of ‘tarpaulin’. Each block is 20 m×20 m in size.

A model of the Stennis Space Center fixed-edge target is seen in the bottom left, in Figure C.4. This target is 20 m×40 m in size. This target consists of two sets of black

and white bar pairs. The pairs are arranged so that one pair lies along the track and the other across the track. The dimensions of each of the bars in the target are 10 m×20 m. The polygons in the fixed edge target are assigned the radiometric properties of black and white paint.

At both the Stennis Space Center and Sjöklilla test sites, the targets have been placed on concrete. The polygons surrounding the models of these test targets have, therefore, been assigned the radiometric properties of concrete.

C.2.2 Radial edge test target: Siemens star

A radial edge target is designed to be used to generate the sensor and atmospheric MTF curves, for spatial evaluation. Figure C.5 shows the Siemens star test target as it appears in the simulation environment.

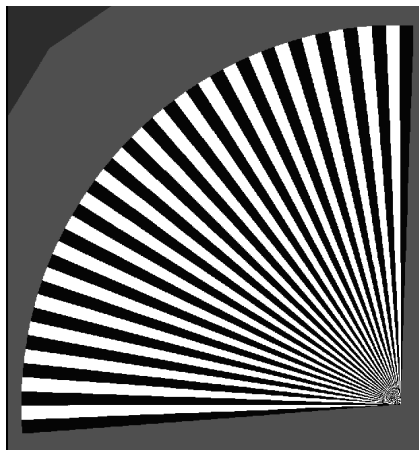


Figure C.5: The radial edge Siemens star target. This target is used for spatial characterisation.

The Siemens star has a 130 m radius and consists of 45 sectors. Each sector has a width of approximately 4.86 m that tapers down towards the centre of the star. The star has an arc angle of 96.39°. The radiometric properties of black and white paint are assigned to the polygons that make up the sectors in the star. The polygons surrounding the Siemens star are assigned the radiometric properties of tree foliage and soil. This is done so that the radiometric properties of the target and surrounding landscape match that of the Siemens star at the Stennis Space Center.

C.2.3 Large reflective surface reference test target

A large reflective surface reference target is designed to evaluate the radiometric values calculated in a simulation. The large reflective surface reference target is modelled after

the Paardefontein calibration site target. Figure C.6 shows the surface reflectance test target as it appears in the simulation environment.

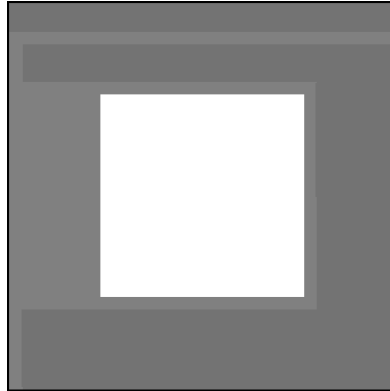


Figure C.6: The Paardefontein calibration and validation site. This target is used to validate the radiometric output of the sensor.

The large surface target is a $100\text{ m} \times 100\text{ m}$ target. The large surface reflectance target is assigned the radiometric properties of grey brick. In the panchromatic band the target has a reflectance of 60%. The surrounding area in the target is assigned the radiometric properties of dry grass and soil. This is done so that the model represents the target and surrounding landscape at the Paardefontein calibration site.

C.2.4 Narrow rectangular line test target

A narrow rectangular line target is designed to evaluate the geometry of the detector layout and the SNR of the system. Figure C.7 shows a section of the narrow rectangular line test targets as they appear in the simulation environment.



(a) Rectangular line target.



(b) Striped rectangular line target.

Figure C.7: The long narrow rectangular line. This target is used to validate the geometry of the detector layout.

The line target in Figure C.7a is used to test the detector layout geometry. The sensor's configuration causes the line target to appear shifted and staggered in the output image. The geometric distortion in the target is able to show accurately the number of pixels by which the detectors in the array are shifted and staggered, relative to each other. The target can also be used to confirm that the correct number of DSAs are implemented and that each DSA consists of the correct number of pixels. The rectangular test target

is 20 km long and 40 m wide. The total surface of the rectangular target is set to have the radiometric properties of 95% reflective tarpaulin.

The striped line target in Figure C.7b is used to verify that the SNR of the sensor imaging band is achieved for the specified $NE\Delta\rho$. The target reflectance is set to form light and dark stripes that will achieve a specific SNR for that imaging band.

C.2.5 Triangular test target

A triangular target is designed to be used when confirming that the across-track and along-track scan motion directions are correct. Figure C.8 shows the triangular target as it appears in the simulation environment.

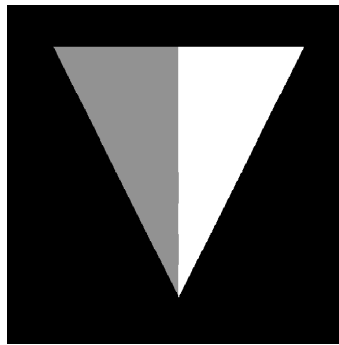


Figure C.8: The triangular target. This target is used to validate the sensor scan direction.

The triangular target has a base length of 25.962 m and a height of 34.616 m. The triangular target appears to point upward or downward in the sensor output image, depending on the along-track scan and/or orbit direction. The left and right sides of the triangular target appear swapped in the sensor output image, depending on the sensor across-track scanning and/or orbit direction. The triangular target is divided into two halves, where each half has a different set of radiometric properties, so that one half of the target has a higher reflectance (appears lighter) than the other. The polygons of the available triangular targets have the radiometric properties of either tree foliage and soil or of 95% and 45% reflective tarpaulin. The shape of the target and the two different reflectivities of the target's surface allow any changes in the scan direction to be noted easily in the sensor output image.

A similar triangular target is mounted on the curved surface of the spherical target.

C.2.6 Spherical test target

A spherical test target is designed to be used when validating the simulated satellite orbit. The sphere target is used to ensure that the sensor is always looking at the surface of a sphere while moving along the calculated orbital trajectory. The surface of the sphere

is patterned with a triangular grid. Any distortion in the pattern in the sensor output image will indicate an error in the satellite's orbit. A triangular target is placed on the curved surface of the sphere in order to confirm that the orbit direction, ascending or descending, is correct.

The spherical target is also used to simulate the geometric distortion in the sensor's output image that is due to a rotating earth. This is done by rotating the spherical target while the satellite sensor images the triangular target on the surface of the sphere.

Figure C.9 shows the spherical and triangular target as it is implemented in the simulation environment.



Figure C.9: The triangle on a sphere. This target is used to validate the orbit's geometry.

The spherical target has a radius of 25.512 km. The triangular target on the curved surface has a base and height length of 3.4 km. The grid has the radiometric properties of dolomite and sea water. These materials are used since the radiometric properties of the materials differ greatly and will thus lead to a high contrast pattern. The high contrast pattern allows errors in the image of the pattern to be noticed easily. The triangular target has the radiometric properties of tree foliage and soil. The remaining polygons surrounding the triangular target, are assigned the radiometric properties of 55% reflective tarpaulin.

A spherical target with a radius equal to that of the earth has been designed. The large target is not used in this study, since a large number of polygons and a large high resolution scene image is required to render the target, which can result in memory errors or a very long simulation time.

Appendix D

Derivation of the Atmospheric Aerosol Point Spread Function

D.1 Aerosol scattering modulation transfer function

The atmospheric aerosol scattering MTF is given as [19], [20],

$$MTF_a(f_r) = \begin{cases} \exp \left\{ - \left(\frac{f_r}{f_c} \right)^2 \sec \theta \int_{h_t}^H [S_a(h, \lambda) + A_a(h, \lambda)] dh \right\}, & \text{if } f_r < f_c \\ \exp \left\{ - \sec \theta \int_{h_t}^H [S_a(h, \lambda) + A_a(h, \lambda)] dh \right\}, & \text{if } f_r \geq f_c. \end{cases} \quad (\text{D.1.1})$$

where λ is the imaging wavelength (m), f_r is the spatial frequency and f_c is the high spatial frequency cut-off, both are expressed as cycles/rad or cycles/m. The spatial frequency cut-off is calculated as,

$$f_c = \frac{r_a}{\lambda f'}, \quad (\text{D.1.2})$$

where r_a is the radius (m) of the scattering particle and f' is the imaging system focal length (m) [19], [20]. Equation (D.1.2) expresses f_c as cycles/m in the focal plane. The spatial cut-off frequency can be expressed in cycles/rad by removing the projection onto the focal plane,

$$f_c = \left(\frac{r_a}{\lambda f'} \times f' \right) = \frac{r_a}{\lambda}. \quad (\text{D.1.3})$$

In (D.1.1), $S_a(h, \lambda)$ and $A_a(h, \lambda)$ are the spectral atmospheric attenuation scattering and absorption coefficients, respectively, with units of m^{-1} . H is the altitude (m) at which the imaging system is located, h_t is the target altitude (m) above the ground and θ is the viewing angle (rad) relative to a vertical path [19], [20].

The sum of $S_a(h, \lambda)$ and $A_a(h, \lambda)$ is expressed as the extinction coefficient (m^{-1}) [1, p. 104],

$$\beta_{\text{ext}}(h, \lambda) = S_a(h, \lambda) + A_a(h, \lambda). \quad (\text{D.1.4})$$

The optical depth along a slanted path through the atmosphere, is then calculated by integrating the extinction coefficient along the path between the target and sensor. The optical depth ($\sigma_{\text{od}}(h, \lambda)$) is unitless and is calculated as [1, pp. 97 and 113],

$$\sigma_{\text{od}}(h, \lambda) = \sec \theta \int_{h_1}^H \beta_{\text{ext}}(h, \lambda) dh. \quad (\text{D.1.5})$$

Equation (D.1.1) is now rewritten in terms of the optical depth as,

$$MTF_a(f_r) = \begin{cases} \exp \left[- \left(\frac{f_r}{f_c} \right)^2 \sigma_{\text{od}}(h, \lambda) \right], & \text{if } f_r < f_c \\ \exp [-\sigma_{\text{od}}(h, \lambda)], & \text{if } f_r \geq f_c. \end{cases} \quad (\text{D.1.6})$$

$MTF_a(f_r)$ is unitless. Listing D.1 shows the implementation of (D.1.6) in Python code. This function is used to plot the MTF curve in Figure 5.4 in Chapter 4.

```
#---Calculate the aerosol scatter MTF
def CalcMtfScatter(f, fc, opticalDepthSecTheta):
    """Calculate the MTF for each spacial frequency
    f is the spatial frequency
    fc = (a/wav)^-1 (cycles/rad)
    - a is the aerosol particle radius (m)
    - wav is the peak wavelength in the imaging spectral band (m)
    opticalDepthSecTheta is the optical depth along the viewing angle
    """

    #--- Set the factor (f/fc)^2. If f>fc then (f/fc)^2 = 1
    factor = np.where((f/(fc*1.0)) < 1.0, (f/(fc*1.0))**2, 1.0)

    #--- Calculate the MTF
    MTFscatter = np.exp(-factor*opticalDepthSecTheta)

    return MTFscatter
```

Listing D.1: Python code snippet used to calculate the aerosol scattering modulation transfer function.

D.2 Aerosol scattering point spread function

The atmospheric aerosol scattering MTF and PSF are circularly symmetric. The atmospheric aerosol scattering PSF is, therefore, derived by taking the inverse Fourier transform of a radial slice through the MTF given in (D.1.6),

$$PSF_a(\omega_r) = \int_{-\infty}^{\infty} MTF_a(f_r) e^{j2\pi\omega_r f_r} df_r. \quad (\text{D.2.1})$$

where ω_r is the angular frequency (rad) and $PSF_a(\omega_r)$ is unitless.

The atmospheric aerosol PSF is divided into a scattered flux ($PSF_{\text{aScatter}}(\omega_r)$) and an unscattered flux ($PSF_{\text{aNoScatter}}(\omega_r)$) PSF component.

The scattered flux PSF is calculated by taking the inverse Fourier transform of the $MTF_a(f_r)$ expression for the frequency range of $f_r < f_c$. The inverse Fourier transform is taken over the user defined frequency range of $-f$ to f , which results in,

$$PSF_{aScatter}(\omega_r) = \int_{-f}^f \exp \left[- \left(\frac{f_r}{f_c} \right)^2 \sigma_{od}(h, \lambda) \right] e^{j2\pi\omega_r f_r} df_r. \quad (D.2.2)$$

From the variables in (D.2.2) a constant x is defined so that,

$$x = \frac{\sigma_{od}(h, \lambda)}{f_c^2} = \sigma_{od}(h, \lambda) \omega_c^2, \quad (D.2.3)$$

with units of (rad). ω_c is the angular cut-off frequency (rad). Equation (D.2.2) is rewritten using the constant x ,

$$PSF_{aScatter}(\omega_r) = \int_{-f}^f e^{-x f_r^2} e^{j2\pi\omega_r f_r} df_r. \quad (D.2.4)$$

Equation (D.2.4) is solved by first finding the derivative of the PSF,

$$\frac{dPSF_{aScatter}(\omega_r)}{d\omega_r} = \int_{-f}^f e^{-x f_r^2} (j2\pi f_r) e^{j2\pi\omega_r f_r} df_r, \quad (D.2.5)$$

and then using integration by parts,

$$\begin{aligned} u &= e^{j2\pi\omega_r f_r} & dv &= j2\pi f_r e^{-x f_r^2} df_r \\ du &= j2\pi\omega_r e^{j2\pi\omega_r f_r} df_r & v &= \frac{-j\pi}{x} e^{-x f_r^2}, \\ \therefore \frac{dPSF_{aScatter}(\omega_r)}{d\omega_r} &= \left[uv \Big|_{-f}^f - \int_{-f}^f v du \right] \\ &= \left[e^{j2\pi\omega_r f_r} \frac{-j\pi}{x} e^{-x f_r^2} \Big|_{-f}^f - \int_{-f}^f \frac{-j\pi}{x} e^{-x f_r^2} j2\pi\omega_r e^{j2\pi\omega_r f_r} df_r \right]. \end{aligned} \quad (D.2.6)$$

Equation (D.2.6) is solved using (D.2.4) and Euler's formula,

$$\begin{aligned} \frac{dPSF_{aScatter}(\omega_r)}{d\omega_r} &= \left[e^{j2\pi\omega_r f_r} \frac{-j\pi}{x} e^{-x f_r^2} \Big|_{-f}^f - \int_{-f}^f \frac{-j\pi}{x} e^{-x f_r^2} j2\pi\omega_r e^{j2\pi\omega_r f_r} df_r \right] \\ &= \frac{-j\pi}{x} e^{-x f^2} (e^{j2\pi\omega_r f} - e^{-j2\pi\omega_r f}) - \frac{2\pi^2\omega_r}{x} \int_{-f}^f e^{-x f_r^2} e^{j2\pi\omega_r f_r} df_r \\ &= \frac{-j\pi}{x} e^{-x f^2} (e^{j2\pi\omega_r f} - e^{-j2\pi\omega_r f}) - \frac{2\pi^2\omega_r}{x} PSF_{aScatter}(\omega_r) \\ &= \frac{-j\pi}{x} e^{-x f^2} \{ [\cos(2\pi\omega_r f) + j \sin(2\pi\omega_r f)] - [\cos(2\pi\omega_r f) - j \sin(2\pi\omega_r f)] \} \\ &\quad - \frac{2\pi^2\omega_r}{x} PSF_{aScatter}(\omega_r) \\ &= \frac{-j\pi}{x} e^{-x f^2} [2j \sin(2\pi\omega_r f)] - \frac{2\pi^2\omega_r}{x} PSF_{aScatter}(\omega_r) \\ \text{let } f &\rightarrow \infty \\ &= - \frac{2\pi^2\omega_r}{x} PSF_{aScatter}(\omega_r). \end{aligned} \quad (D.2.7)$$

Equation (D.2.7) is recognisable as a separable differential equation. The differential equation is solved as follows,

$$\begin{aligned}
 \frac{dPSF_{\text{aScatter}}(\omega_r)}{d\omega_r} &= -\frac{2\pi^2\omega_r}{x} PSF_{\text{aScatter}}(\omega_r) \\
 \therefore \frac{dPSF_{\text{aScatter}}(\omega_r)}{PSF_{\text{aScatter}}(\omega_r)} &= -\frac{2\pi^2\omega_r}{x} d\omega_r \\
 \therefore \int \frac{dPSF_{\text{aScatter}}(\omega_r)}{PSF_{\text{aScatter}}(\omega_r)} &= -\int \frac{2\pi^2\omega_r}{x} d\omega_r \\
 \therefore \ln [PSF_{\text{aScatter}}(\omega_r)] &= -\frac{2\pi^2\omega_r^2}{2x} + C \\
 \therefore \exp \{ \ln [PSF_{\text{aScatter}}(\omega_r)] \} &= \exp \left(-\frac{\pi^2\omega_r^2}{x} + C \right) \\
 \therefore PSF_{\text{aScatter}}(\omega_r) &= \exp(C) \exp \left(-\frac{\pi^2\omega_r^2}{x} \right) \\
 &= A \exp \left(-\frac{\pi^2\omega_r^2}{x} \right). \tag{D.2.8}
 \end{aligned}$$

ω_r , in (D.2.8), is set to zero in order to calculate the value of the constant $A = \exp(C)$,

$$PSF_{\text{aScatter}}(0) = A. \tag{D.2.9}$$

ω_r , in (D.2.4), is also set to zero which results in the equation being a Gaussian integral, where $f \rightarrow \infty$ [79],

$$\begin{aligned}
 PSF_{\text{aScatter}}(0) &= \int_{-f}^f e^{-xf_r^2} e^{-j2\pi 0 f_r} df_r \\
 &= \int_{-f}^f e^{-xf_r^2} df_r \\
 &= \sqrt{\frac{\pi}{x}}. \tag{D.2.10}
 \end{aligned}$$

The aerosol scatter PSF is then calculated by substituting (D.2.10) into (D.2.8) where the value of x is calculated using (D.2.3),

$$P_{\text{aScatter}}(\omega_r) = \sqrt{\frac{\pi}{x}} \exp \left(-\frac{\pi^2\omega_r^2}{x} \right). \tag{D.2.11}$$

The unscattered radiation PSF is calculated by taking the inverse Fourier transform of the $MTF_{\text{a}}(f_r)$ expression for the frequency range of $f_r \geq f_c$. The inverse Fourier transform is taken over the user defined frequency range of $-f$ to f , which results in,

$$\begin{aligned}
 P_{\text{aNoScatter}}(\omega_r) &= \int_{-f}^f e^{-\sigma_{\text{od}}(h,\lambda)} e^{j2\pi\omega_r f_r} df_r \\
 &= e^{-\sigma_{\text{od}}(h,\lambda)} \int_{-f}^f e^{j2\pi\omega_r f_r} df_r \\
 &= e^{-\sigma_{\text{od}}(h,\lambda)} \delta(\omega_r), \tag{D.2.12}
 \end{aligned}$$

where the $\delta(\omega_r)$ is the Dirac delta function. The Dirac delta function is described as being zero for all ω_r values except at $\omega_r = 0$ rad, where it has an infinitely large value. The Dirac delta function results in $P_{\text{aNoScatter}}(\omega_r)$ having a non-zero value at only the centre pixel. The Dirac delta function is approximated by the Gaussian function [80],

$$\delta(\omega_r) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(\omega_r - \mu)^2}{2\sigma^2}\right]. \quad (\text{D.2.13})$$

The Gaussian function has a mean value of μ (rad) and a standard deviation of σ (rad). The mean value is set to zero and the standard deviation is set so that the full width at half maximum (FWHM) value is equal to a single pixel width. The standard deviation is calculated in terms of the FWHM by setting the Gaussian curve in (D.2.13) equal to the FWHM value and solving for σ . The constant factor of $\frac{1}{\sigma\sqrt{2\pi}}$ can be ignored and the FWHM point is at half of the $\delta(\omega_r)$ peak. The peak is at the frequency $\omega_r = 0$ rad. The derivation is shown in (D.2.14) [80],

$$\exp\left[-\frac{(\omega_r - \mu)^2}{2\sigma^2}\right] = \frac{1}{2} \delta(0)$$

setting $\mu = 0$ rad and finding the natural logarithm,

$$\therefore -\frac{(\omega_r)^2}{2\sigma^2} = -\ln(2)$$

$$\therefore \omega_r = \sigma\sqrt{2\ln(2)}$$

the FWHM is from $-\omega_r$ to ω_r , which results in a width of $2\omega_r$,

$$\therefore \text{FWHM} = 2\sigma\sqrt{2\ln(2)}$$

$$\therefore \sigma = \frac{\text{FWHM}}{2\sqrt{2\ln(2)}}. \quad (\text{D.2.14})$$

The atmospheric aerosol PSF is therefore defined as,

$$PSF_{\text{a}}(\omega_r) = \begin{cases} \sqrt{\frac{\pi}{x}} \exp\left(-\frac{\pi^2\omega_r^2}{x}\right), & \text{if } |\omega_r| > \frac{\omega_{\text{pixelPitch}}}{2} \\ \exp[-\sigma_{\text{od}}(h, \lambda)] \delta(\omega_r), & \text{if } |\omega_r| \leq \frac{\omega_{\text{pixelPitch}}}{2}, \end{cases} \quad (\text{D.2.15})$$

where $\omega_{\text{pixelPitch}}$ is the angular frequency (rad) for a single pixel. The PSF is calculated for an odd number of pixels so that the PSF will have a centre pixel. Only the centre pixel has the unscattered PSF component, therefore, the PSF angular frequency conditions use the $\frac{\omega_{\text{pixelPitch}}}{2}$ term.

Listing D.2 shows the implementation of (D.2.15) in Python code. This function is used to plot the PSF curve in Figure 5.5 in Chapter 4.


```

#---PSF of the scattered MTF
def CalcPsfScatter(w, wc, opticalDepthSecTheta, pixelPitch, fl):
    """Calculate the aerosol scattering point spread function.

    w is the angular frequency (w=1/f) (rad/cycles = rad)
    wc is the angular cut-off frequency (rad/cycles = rad)
    opticalDepthSecTheta is the MODTRAN tape 7 optical depth at
        a wavelength (unitless)
    pixelPitch is the detector pitch (m)
    fl is the focal length (m)
    """

    #--- Calculate the constant factor in the PSF
    # Units: unitless*(rad/cycles)^2 -> rad^2
    x = (opticalDepthSecTheta*(wc**2))

    #--- Set the impulse response mu and sigma values
    mu = 0.0 #Peak at the centre pixel
    #Set FWHM to be 1 pixel IFOV (rad)
    FWHM = 1.0*(pixelPitch/fl) #Centre peak is only one pixel wide
    sigma = FWHM/(2.0*np.sqrt(2*np.log(2)))

    #--- Calculate the impulse response
    impulse = gaussian(w, sigma, mu)

    #--- Calculate the scatter PSF
    #The impulse responses should only be used for the centre pixel
    #(width of one pixel FOV = FWHM)
    # Units case1: sqrt(pi)/(sqrt(opticalDepthSecTheta)*wc) * exp(-(w**2)/(4*x))
    # = rad/(unitless*rad)* (rad^2/rad^2) -> unitless
    # Units case2: exp(-opticalDepthSecTheta)*impulse)
    # = unitless*unitless = unitless
    PSFscatter = np.where( np.abs(w)>(FWHM/2.0),
                           (np.sqrt(np.pi/x) * np.exp(-((np.pi**2)*(w**2))/x)),
                           (np.exp(-opticalDepthSecTheta)*impulse))

    return PSFscatter

```

Listing D.2: Python code snippet used to calculate the aerosol scattering point spread function.