ERADIATE

An open-source radiative transfer model for the Earth observation community

Vincent Leroy & The Eradiate Team

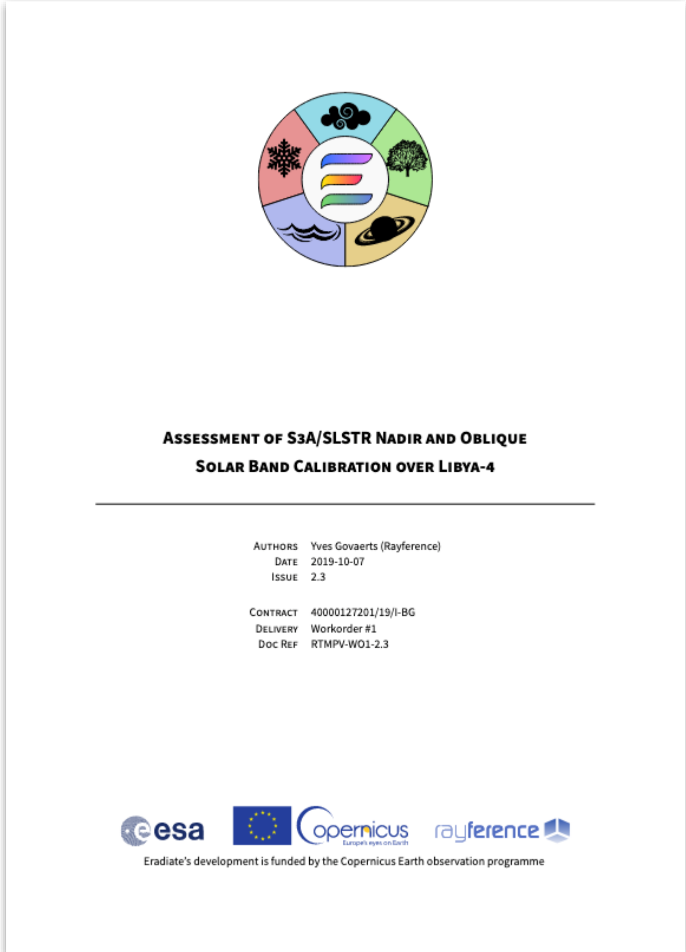
Rayference

RAMI workshop 2023 // Varese // June 7th, 2023

Instruments are accurate, our models don’t follow

- Radiometric calibration accuracy can now get close to 1% with some satellite-borne instruments
- RTMs used to verify calibration are not accurate enough
  - Lack of 3D features
    1D plane-parallel assumption: Earth is flat, no 3D surface features
  - Missing molecular absorption modelling features
    Lack of chemical profile update

Goal: high accuracy suitable for such use cases



Assessment of S3A/SLSTR Nadir and Oblique Solar Band Calibration over Libya-4

Authors Yves Govaerts (Rayference)
Date 2019-10-07
Issue 2.3

Contract 40000127201/19/I-BG
Delivery Workorder #1
Doc Ref RTMPV-WO1-2.3

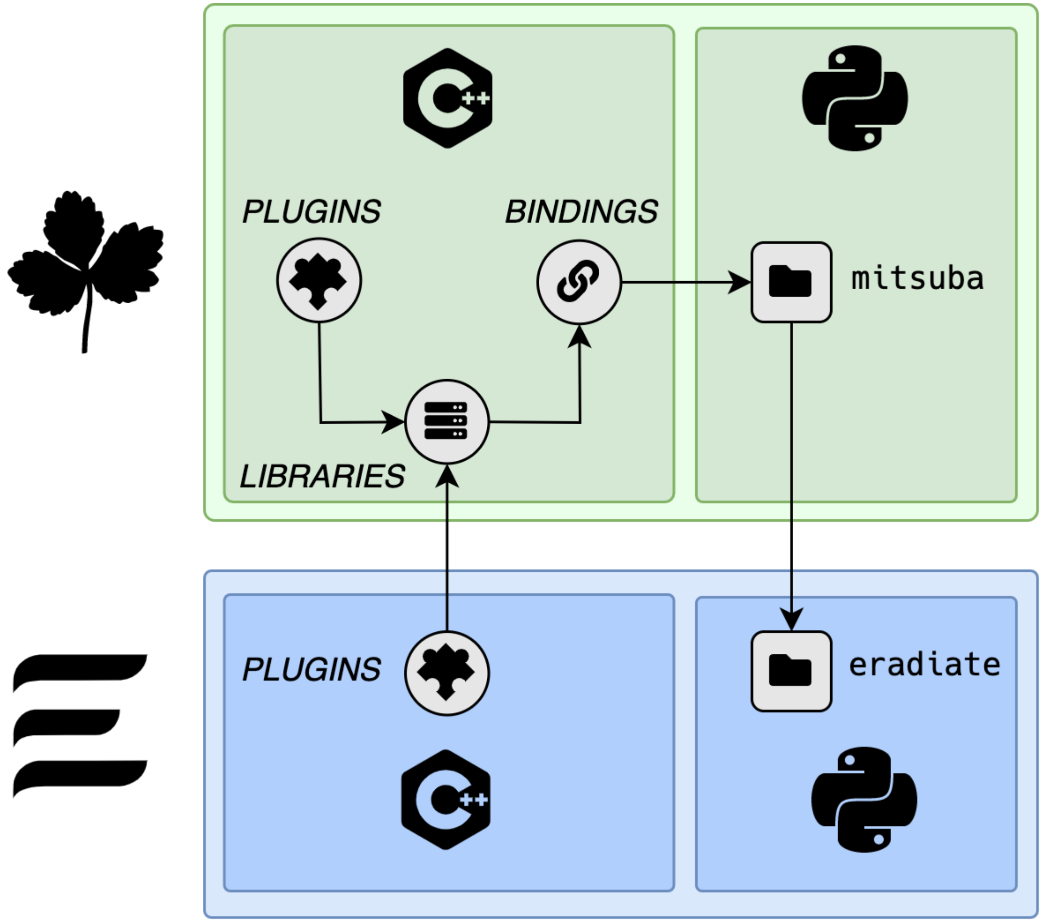
Eradiate’s development is funded by the Copernicus Earth observation programme

Govaerts (2019), work order for Eradiate Development Phase 1

Goals and scope

- **General-purpose 3D RTM for cal/val**
  - Common platform to integrate advances from all subcommunities
  - Prioritize accuracy when selecting numerical methods and algorithms
  - Extensive, reproducible and systematic testing
- **Researched and traceable data and algorithms**
  - Get to the bottom of the state of the art, document it for the benefit of the community
- **Designed for interactive analysis**
  - Python interface
    $\Rightarrow$ integration in interactive computing environments
  - Modern data processing chain
    $\Rightarrow$ data-centric workflow based on NetCDF/xarray
- **Community-oriented**
  - Extensive documentation (user and dev guides, tutorials, API docs)
  - Open-source, hosted publicly
  - Involve community (issues, pull requests, workshops)

Architecture overview



Radiometric kernel:
Mitsuba 3 rendering system

- Retargetable design
- Plugin architecture
- Great Python bindings
- Great code quality

Specific plugins

- Scattering models
- Sensors
- Volume data source

Python package

- Scene generator and pre-processor
- Simulation runner
- Post-processing pipeline

Feature overview

Fundamentals: scalar 1D radiative transfer model

Surface

- Smooth surface (Lambertian, RPV, tabulated; more to come)

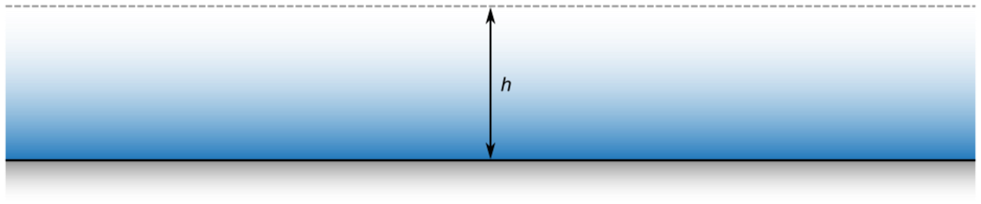
Atmosphere

- 1D atmospheric model
- Molecular component (standard AFGL 1986 profiles in CKD mode, custom profiles in CKD and LBL modes)
  ⇒ Any profile can be used (*e.g*. CAMS)
- Arbitrary number of particle components (parametrized by OT, albedo and phase function)

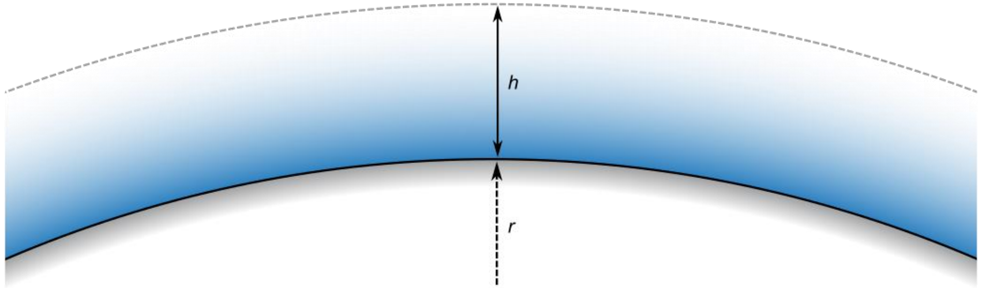
Fundamentals: scalar 1D radiative transfer model

Scene geometry

- Plane-parallel (PPG) or spherical-shell (SSG) geometries
- PPG as a reference
- Only SSG ensures corrects radiance estimates at grazing angles



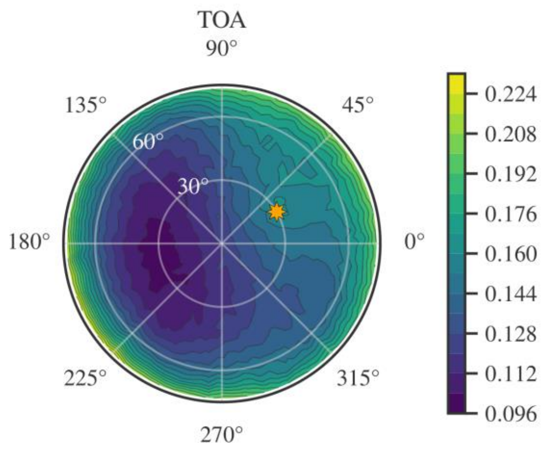
vs

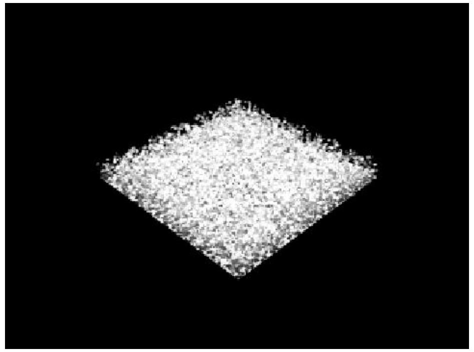


Fundamentals: scalar 1D radiative transfer model

Measure models

- TOA radiance
  - TOA BRF accessible without additional computation (irradiance is known)
- In situ radiance
  - In situ HDRF accessible with additional computation ⇒ higher variance
- TOA exitance (reflected flux)
  - Total albedo accessible without additional computation
- Perspective camera
- Library of spectral response functions + bring your own





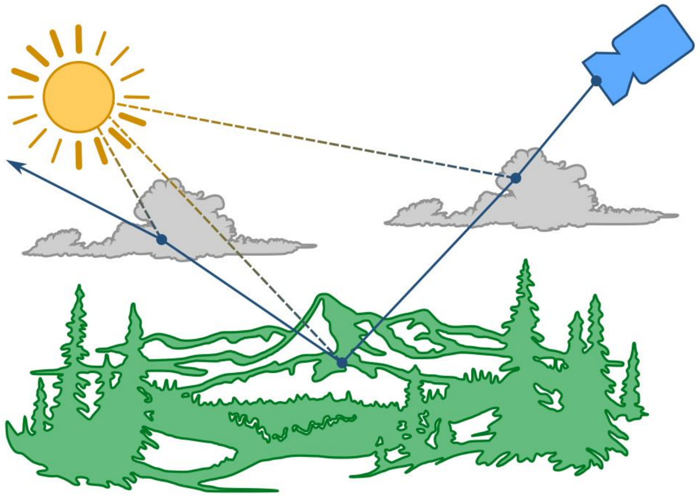
Fundamentals: scalar 1D radiative transfer model

Illumination

- Collimated (ideal)
- Upcoming finite-sized solar model
- Many solar irradiance spectra to choose from + bring your own

Integrator

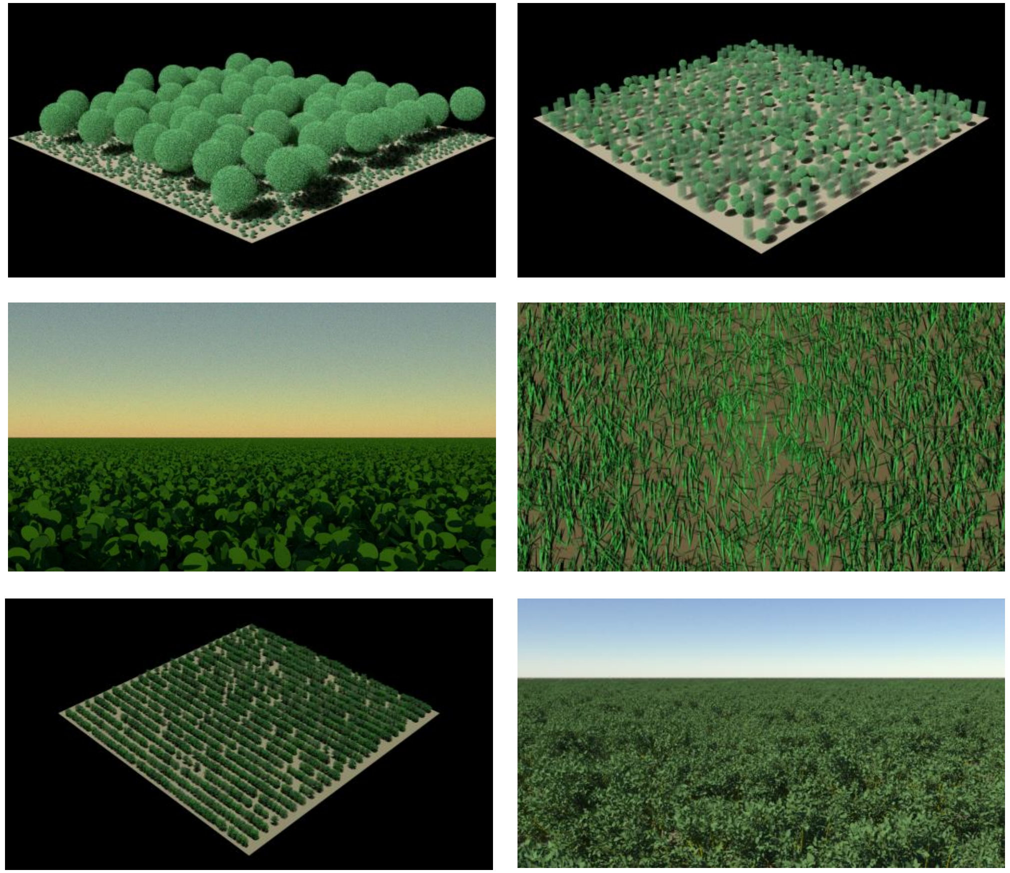
- Stock Mitsuba volumetric path tracer (backward algorithm)
  - Next-event estimation + Russian roulette
  - Null-collision-based free-flight distance sampling



3D surface representation

Explicit 3D vegetation

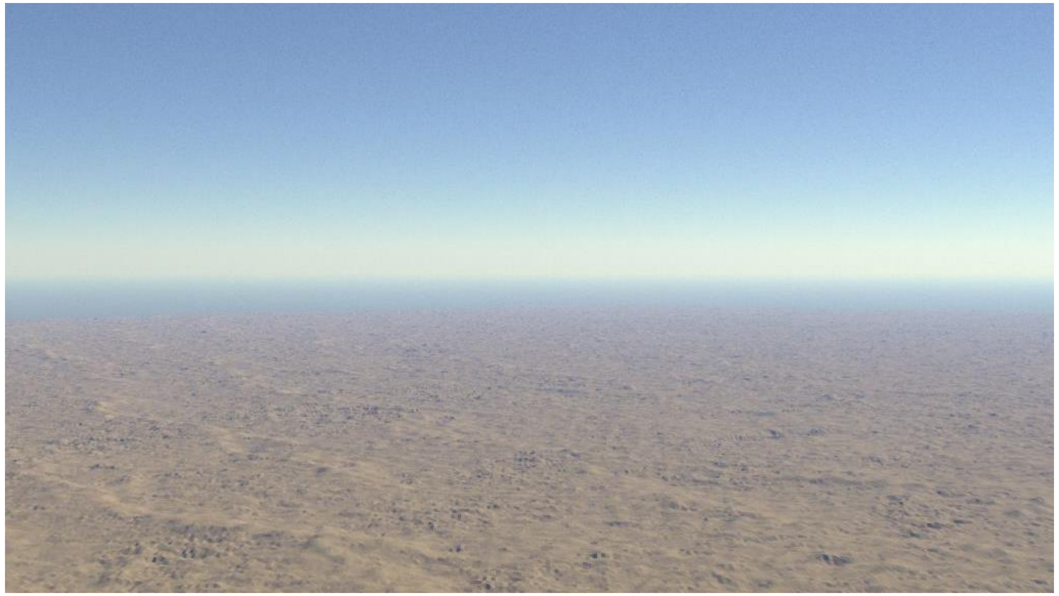
- Abstract disk-based canopy builder
- Arbitrary mesh-based canopy models
- Bilambertian leaf scattering model
- Unit cell can be padded with clones
  ⇒ pseudo-periodic geometry

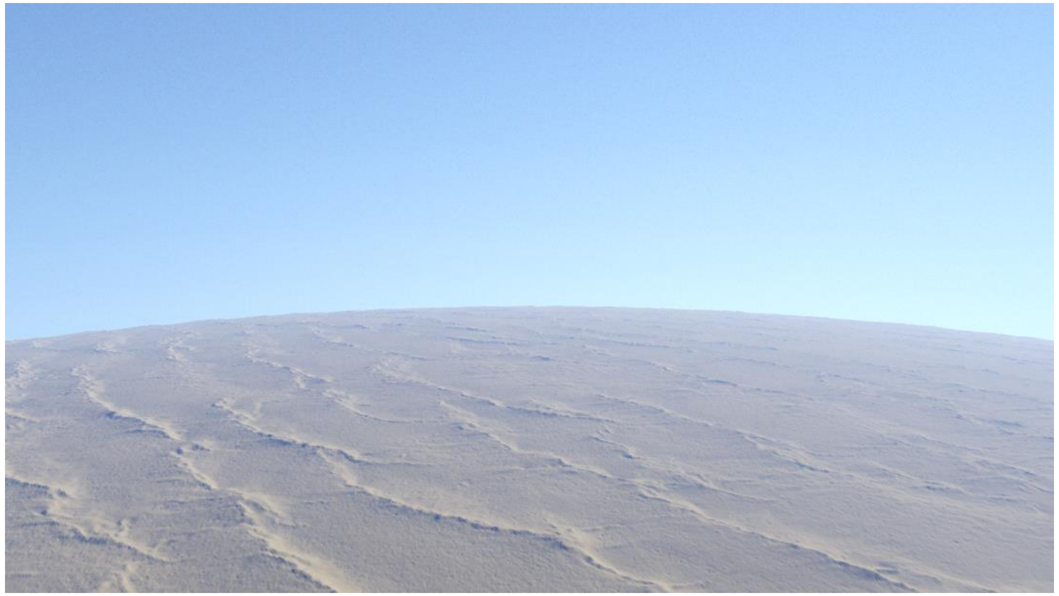


3D surface representation

Digital elevation model

- Automated DEM data tessellation from NetCDF elevation file
  ⇒ Any data source can be used (*e.g*. Copernicus)
- PPG and SSG support





User interface

- Based on Python programming
- Designed for usage in interactive Python session (*e.g*. Jupyter Notebook)
- Fully scriptable ⇒ easy to integrate in Python scientific workflow

```python
import eradiate.scenes as ertsc
import eradiate.experiments as ertxp

exp = ertxp.OneDimExperiment(
    surface=ertsc.bsdfs.RPVBSDF(),
    atmosphere=ertsc.atmosphere.MolecularAtmosphere.afgl_1986(),
    illumination=ertsc.illumination.DirectionalIllumination(
        zenith=15.0,
        azimuth=0.0,
    ),
    measures=ertsc.measure.MultiDistantMeasure.from_viewing_angles(
        id="toa_brf",
        zeniths=np.arange(-75, 76, 5),  # Cover the [-75°, 75°] range with 5°
        azimuths=0,  # Same value as SAA to cover the principal plane
        spectral_cfg={
            "srf": "sentinel_2a-msi-5",
            "bin_set": "10nm",
        },
        spp=1000,
    ),
)
```

```python
[7]: results = eradiate.run(exp)
     results
```

Spectral loop [710:15]: 48/48 00:05, ETA=00:00

[7]: xarray.Dataset

▸ Dimensions: (**sza**: 1, **saa**: 1, **w**: 3, **y_index**: 1, **x_index**: 31, **srf_w**: 22)

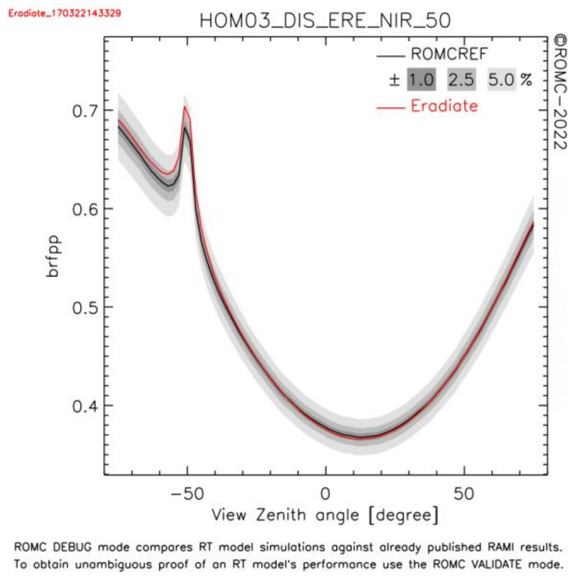
▸ Coordinates: (13)

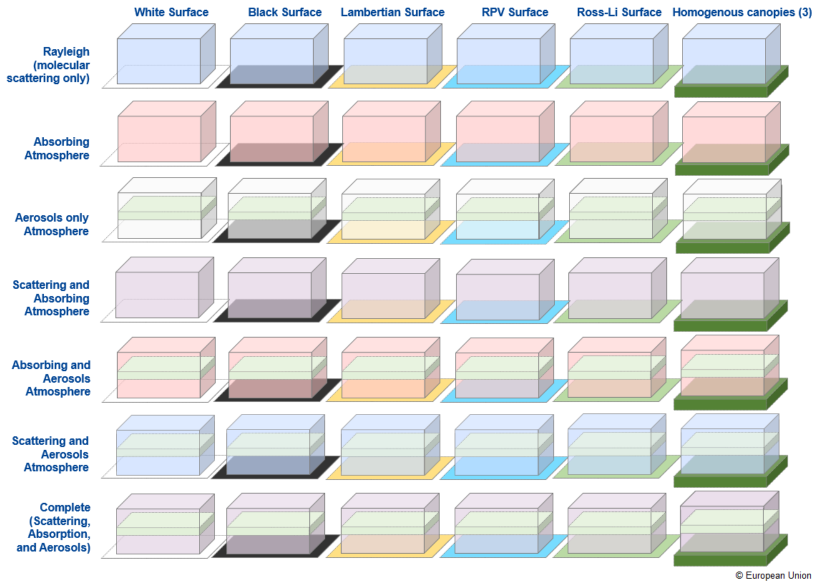
▾ Data variables:

radiance	(sza, saa, w, y_index, x_index)	float64	0.09475 0.0949 ......
spp	(sza, saa, w)	float64	1e+03 1e+03 1e+03
irradiance	(sza, saa, w)	float64	1.412 1.39 1.349
srf	(srf_w)	float64	0.0 0.02836 0.123...
brdf	(sza, saa, w, y_index, x_index)	float64	0.06711 0.06722

Testing & validation

- Unit tests
- System tests with regression detection (improvement in progress)
- Benchmarking: ROMC / RAMI-V / RAMI4ATM
  To be extended (*e.g.* IPRT for polarization components)





Application: UAV reflectance measurement simulation and exploration

S. Schunke, V. Leroy, Y. Govaerts

Background: surface reflectance measurements

Reflectance measurements are used to characterize surfaces.

Many measurement methods exist, *e.g.*:

- Gonioreflectometer
- Tower-mounted radiometer
- **Airborne radiometer (typically on a UAV)**

Latini et al. (2021)

DOI: 10.1109/IGARSS47720.2021.9554496

How does one measure reflectance with a UAV?

Did you say “reflectance”?

Problem: “reflectance” is unspecific

TABLE 1. *Proposed nomenclature for nine kinds of reflectance**

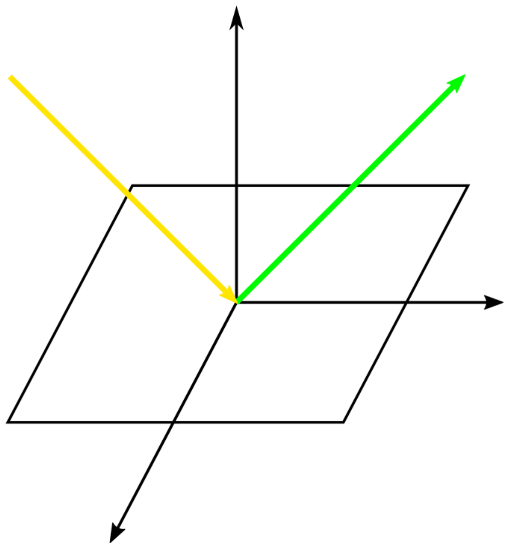
1. Bidirectional reflectance	$d\rho(\theta_i,\phi_i;\theta_r,\phi_r)$	
2. Directional-conical reflectance[a]	$\rho(\theta_i,\phi_i;\omega_r)$	$= \int_{\omega_r} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r$
3. Directional-hemispherical reflectance	$\rho(\theta_i,\phi_i;2\pi)$	$= \int_{2\pi} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r$
4. Conical-directional reflectance[a]	$d\rho(\omega_i;\theta_r,\phi_r)$	$= (d\Omega_r/\Omega_i)\cdot\int_{\omega_i} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_i$
5. Biconical reflectance[a]	$\rho(\omega_i;\omega_r)$	$= (1/\Omega_i)\cdot\int_{\omega_i}\int_{\omega_r} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r\cdot d\Omega_i$
6. Conical-hemispherical reflectance[a]	$\rho(\omega_i;2\pi)$	$= (1/\Omega_i)\cdot\int_{\omega_i}\int_{2\pi} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r\cdot d\Omega_i$
7. Hemispherical-directional reflectance	$d\rho(2\pi;\theta_r,\phi_r)$	$= (d\Omega_r/\pi)\cdot\int_{2\pi} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_i$
8. Hemispherical-conical reflectance[a]	$\rho(2\pi;\omega_r)$	$= (1/\pi)\cdot\int_{2\pi}\int_{\omega_r} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r\cdot d\Omega_i$
9. Bihemispherical reflectance	$\rho(2\pi;2\pi)$	$= (1/\pi)\cdot\int_{2\pi}\int_{2\pi} f_r(\theta_i,\phi_i;\theta_r,\phi_r)\cdot d\Omega_r\cdot d\Omega_i$

Nicodemus et al. (1977)

Which one?

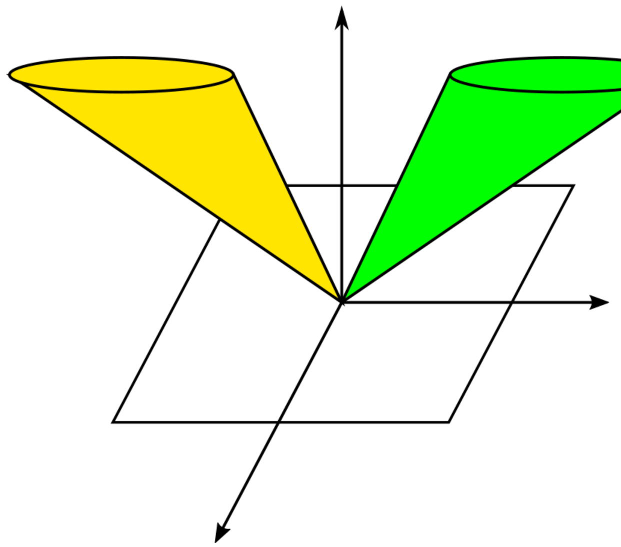
Meet the reflectance family

Intrinsic quantity
(inaccessible experimentally)



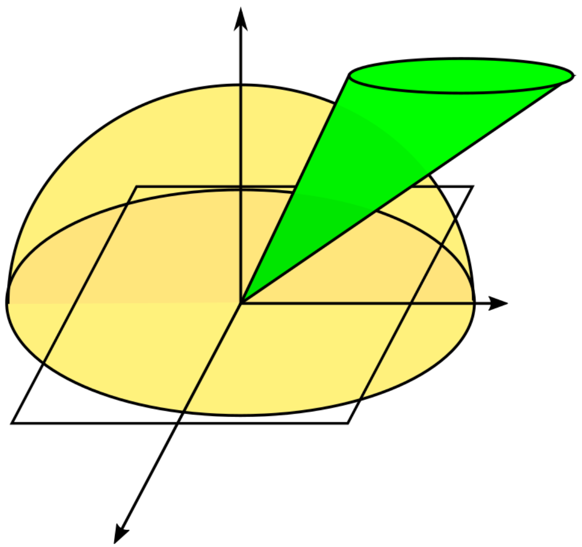
BRF
Bidirectional
reflectance factor

Lab accessible



BCRF
Biconical
reflectance factor

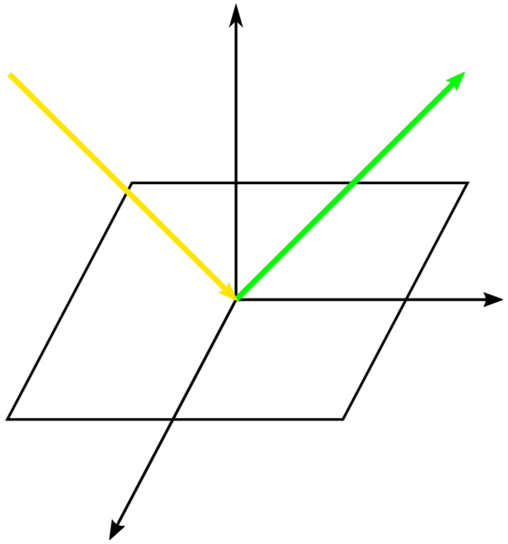
Field measurement



HCRF
Hemispherical-conical
reflectance factor

Meet the reflectance family

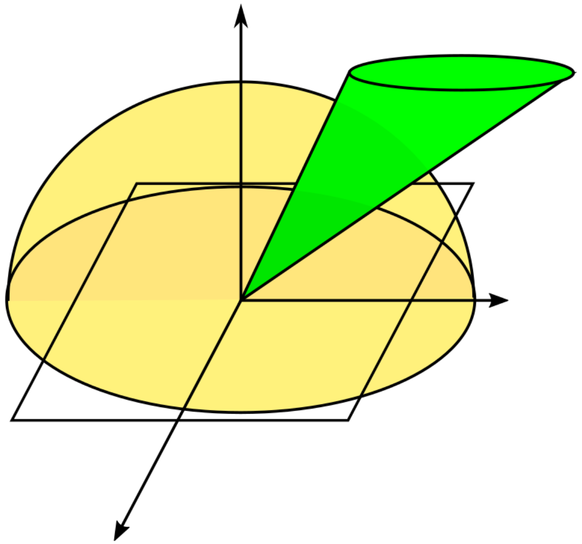
Intrinsic quantity (FRM)
(inaccessible experimentally)



BRF
Bidirectional
reflectance factor

Is HCRF a good proxy for BRF?

Field measurement



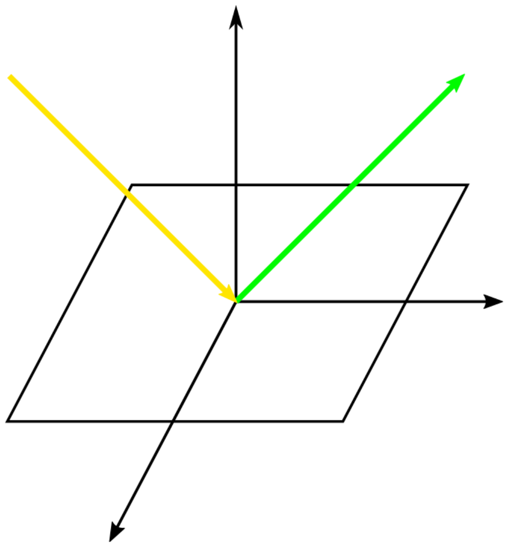
HCRF
Hemispherical-conical
reflectance factor

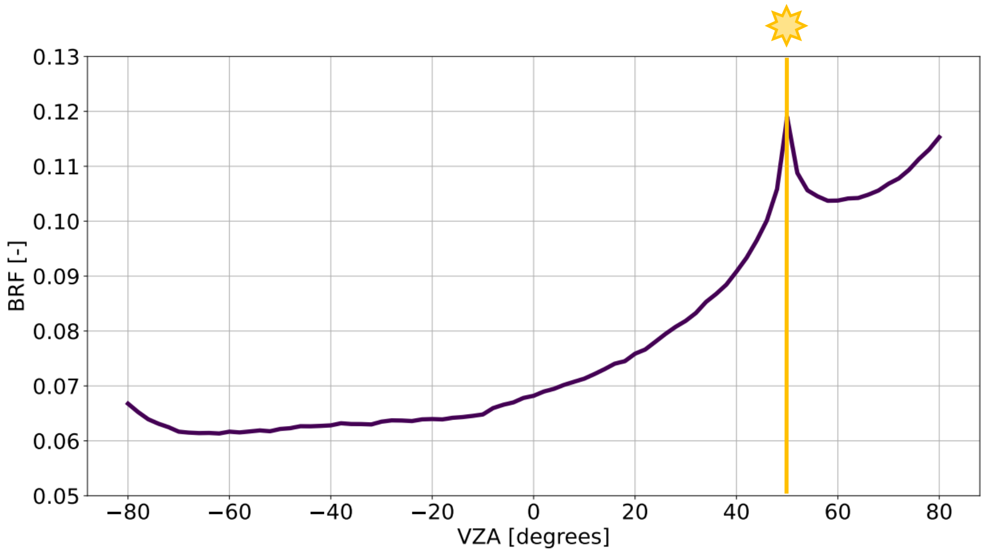
Test scene: a simple grass field

- Grass model provides BRF with hotspot (typical of vegetated covers)
- 1D plane parallel atmosphere, monochromatic simulations at 550nm

Reference case

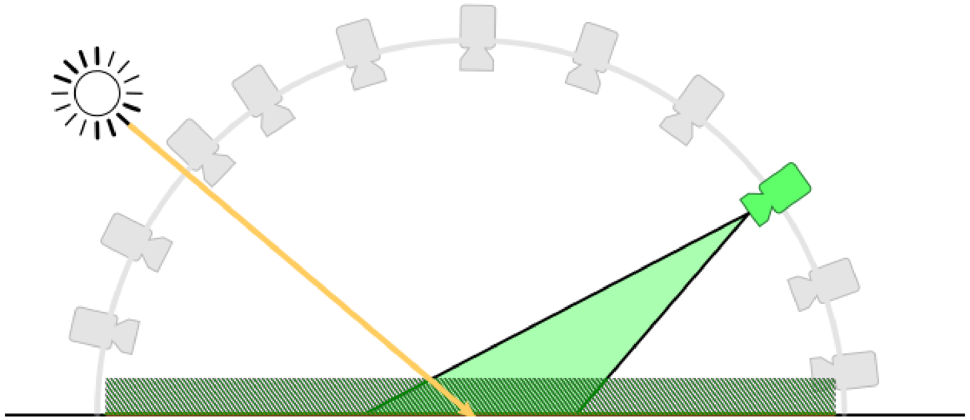
Simulate the BRF of the vegetated surface
⇒ used as a **reference** hereafter

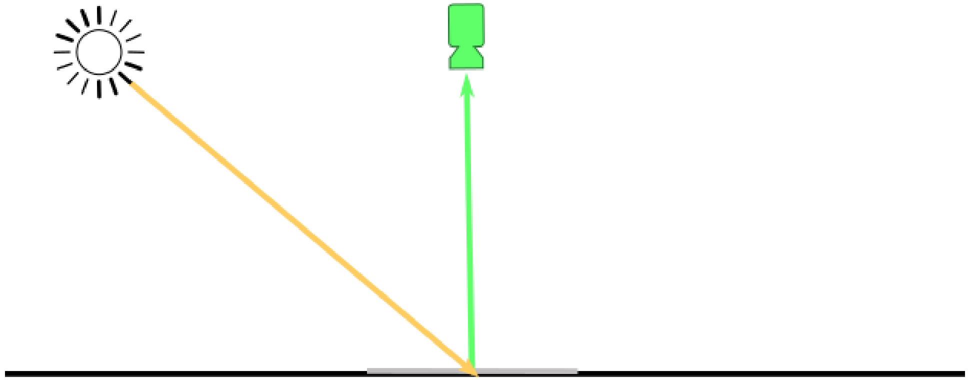




HRDF simulation protocol

1. Record radiance reflected by region of interest (ROI)
2. Calibration: record radiance on calibrated reference panel (CRP)

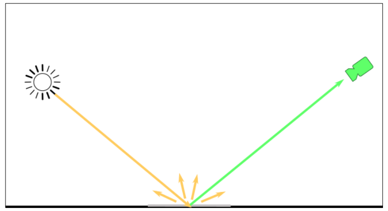




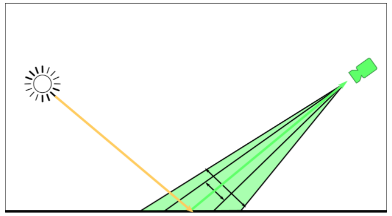
3. $\text{HCRF} = L_{\text{ROI}} / L_{\text{CRP}}$

Correct only if CRP is perfectly diffuse and measurement is instantaneous

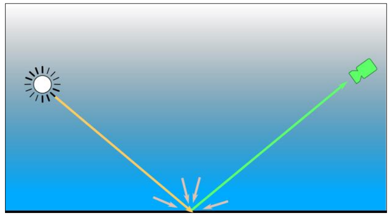
What can make the recorded quantity depart from the BRF?



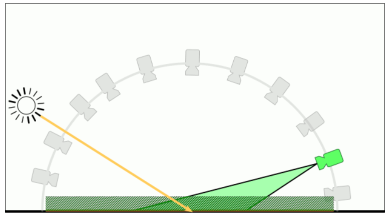
Non Lambertian CRP



Radiometer field of view

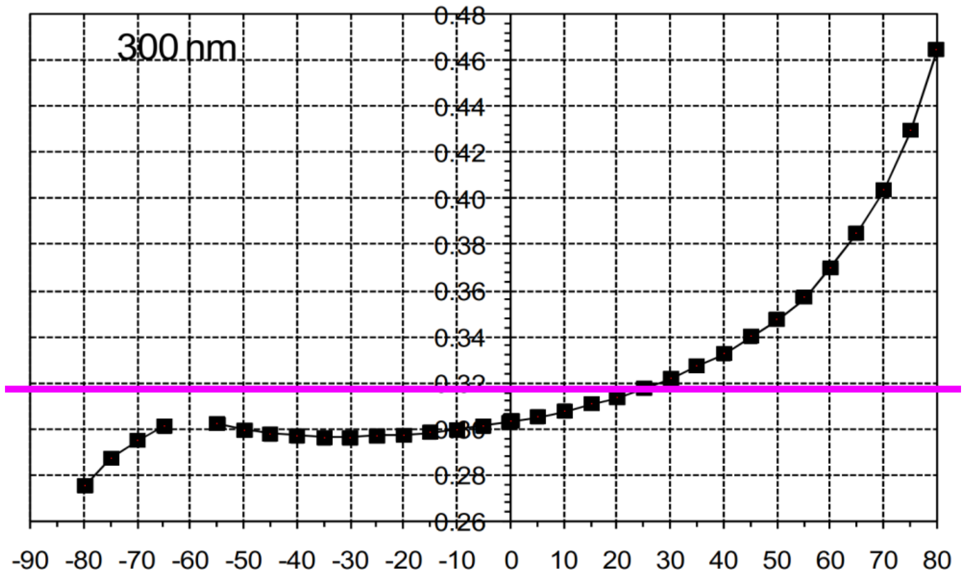


Atmospheric scattering

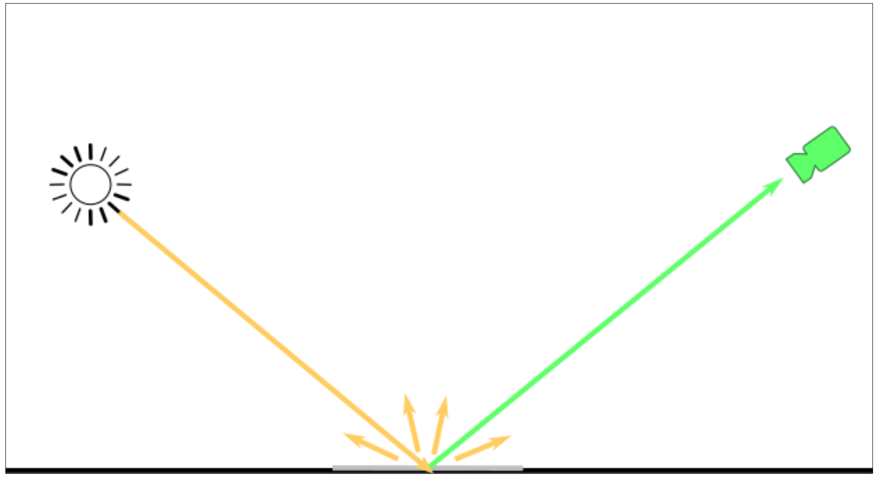


Flight duration

What can make the recorded quantity depart from the BRF?



Georgiev and Butler (2008)
DOI: 10.1117/12.795931



Non Lambertian CRP

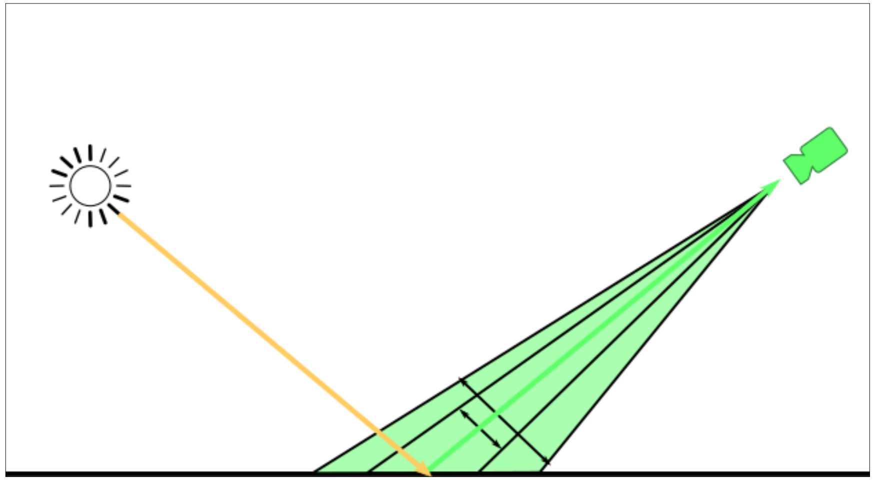
Radiometer field of view

Atmospheric scattering

Flight duration

The CRP is not perfectly diffuse

What can make the recorded quantity depart from the BRF?



Non Lambertian CRP

Radiometer field of view

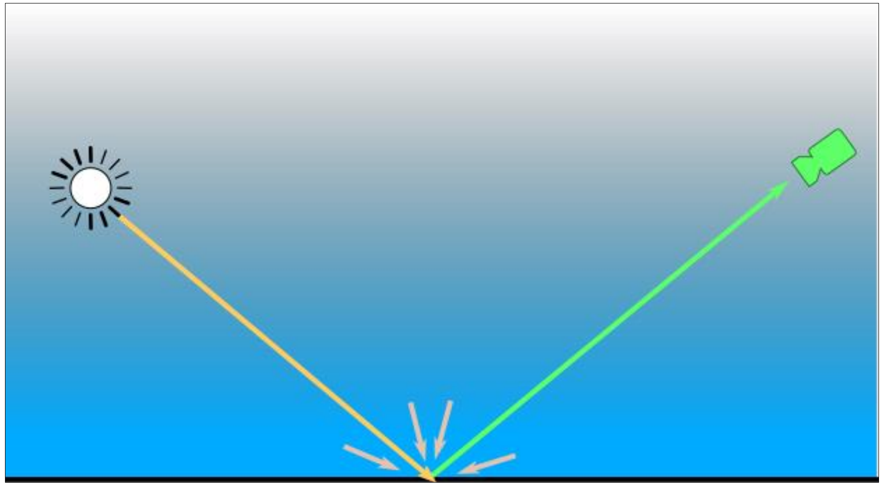
Atmospheric scattering

Flight duration

The sensor is not perfectly directional

The larger the FOV, the less directional the measure

What can make the recorded quantity depart from the BRF?



Non Lambertian CRP

Radiometer field of view

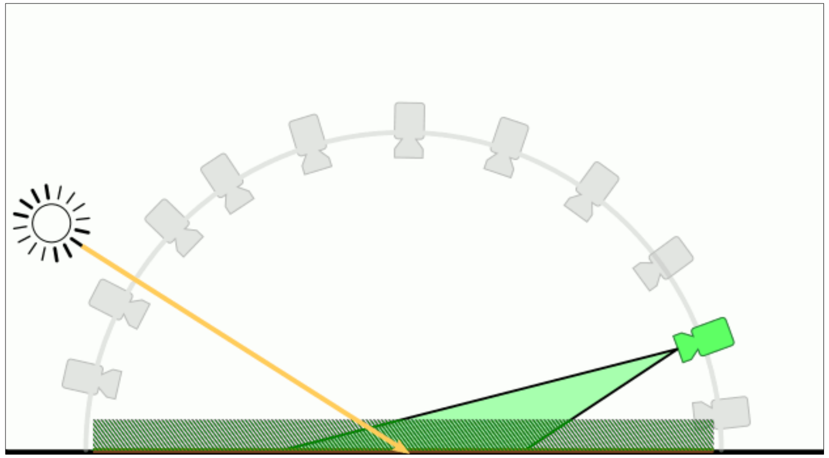
Atmospheric scattering

Flight duration

The illumination is not perfectly directional

The stronger the atmospheric scattering, the less directional the illumination

What can make the recorded quantity depart from the BRF?



Non Lambertian CRP

Radiometer field of view

Atmospheric scattering

Flight duration

Movement of the sun due to **non-zero flight duration**

Acquisition is not instantaneous

How do the effects compare?

Simulation campaign parameters

Parameter	Value 1	Value 2
Calibrated reference panel	Lambertian	Measured BRDF
Field of view	1°	30°
Atmospheric scattering	None	Optical thickness of 0.4
Flight duration	Instantaneous	20 minutes

How do the effects compare?

Simulation campaign results

Parameter	Max rel. deviation
Atmospheric scattering	14%
Field of view	5%
Calibrated reference panel	<1%
Flight duration	<1%

Conclusions

Dominant effects:

- atmospheric scattering ⇒ perform measurements on a clear-sky day
- field of view ⇒ restrict the effective FOV of the instrument as much as possible

Eradiate is FOSS

Hosted on GitHub
⇒ github.com/eradiate/eradiate

- Star and watch for releases
- Get the code
- Reach out and contribute (discussions, issues, PRs)

Documentation on Read The Docs
⇒ eradiate.readthedocs.io

- Tutorials
- Full API documentation