Introduction to Eradiate

ERADIATE *v0.21.1*

V. Leroy

Rayference

Eradiate Preview and Training // 07-12-2021

esa

Copernicus
Europe's eyes on Earth

rayference

A word of warning

- Eradiate is **beta** software as of 07/12/2021
- We are aware of and addressing two important issues which make it unfit for production use
  - Incorrect ozone continuum spectrum in the visible region: we are investigating issues in our datasets
  - Inaccurate shape intersection in certain conditions (notably, coupled atmosphere-canopy simulations): this will be fixed with the integration of a major update of Mitsuba 2

Features [I]

- General use cases
  - In 1D atmosphere
  - Over explicit 3D canopy
  - Both together! (experimental)
- Spectral modes
  - Line-by-line
  - Band (correlated *k*-distribution, 1 nm or 10 nm bins)
- Measures
  - Top-of-atmosphere/canopy radiance/BRF/BRDF
  - Perspective camera
  - Apply instrument spectral response (shipped or custom)
- Illumination
  - Directional (Dirac angular distribution)
  - Selectable irradiance spectrum
  - Account for irradiance variations using ephemeris
- Atmosphere
  - Molecular atmosphere (Rayleigh scattering, absorption)
    - AFGL 1986 thermophysical property profiles (*U.S. Standard*, more coming)
    - Adjustable species concentration of H_2O and O_3
  - Aerosols (particle layers)
    - Any number, parametrised by optical thickness
    - Adjustable density profile

Features [II]

- Written in modern Python (3.7, soon 3.8+) and C++ (17)
  - Mitsuba 2 as radiometric kernel (custom plugins + collaborative development)
  - Use of widespread Python libraries
- Embedded in Python data processing environments
  - Currently deployed with Conda-based development env (ASAP Pip and Conda install)
  - Designed for interactive usage (Jupyter notebook)
  - Data workflow based on xarray

Basic concepts [I]

- The **Experiment** is Eradiate's most general concept. It defines a simulation with natural parameters and covers all aspects:
  - physical object definition and positioning (type, geometry, radiative properties);
  - illumination definition (type, spectrum…);
  - measure definition (type, positioning, angular coverage, spectral configuration, accuracy control…);
  - simulation parametrisation (Monte Carlo algorithms, sampling…).
- **Experiment** used to be called **Solver Application** in previous Eradiate iterations.
- Eradiate currently has 3 **Experiment** classes:
  - `OneDimExperiment`: 1D model;
  - `RamiExperiment`: explicit canopy simulator;
  - `Rami4ATMExperiment`: merges the two previous.

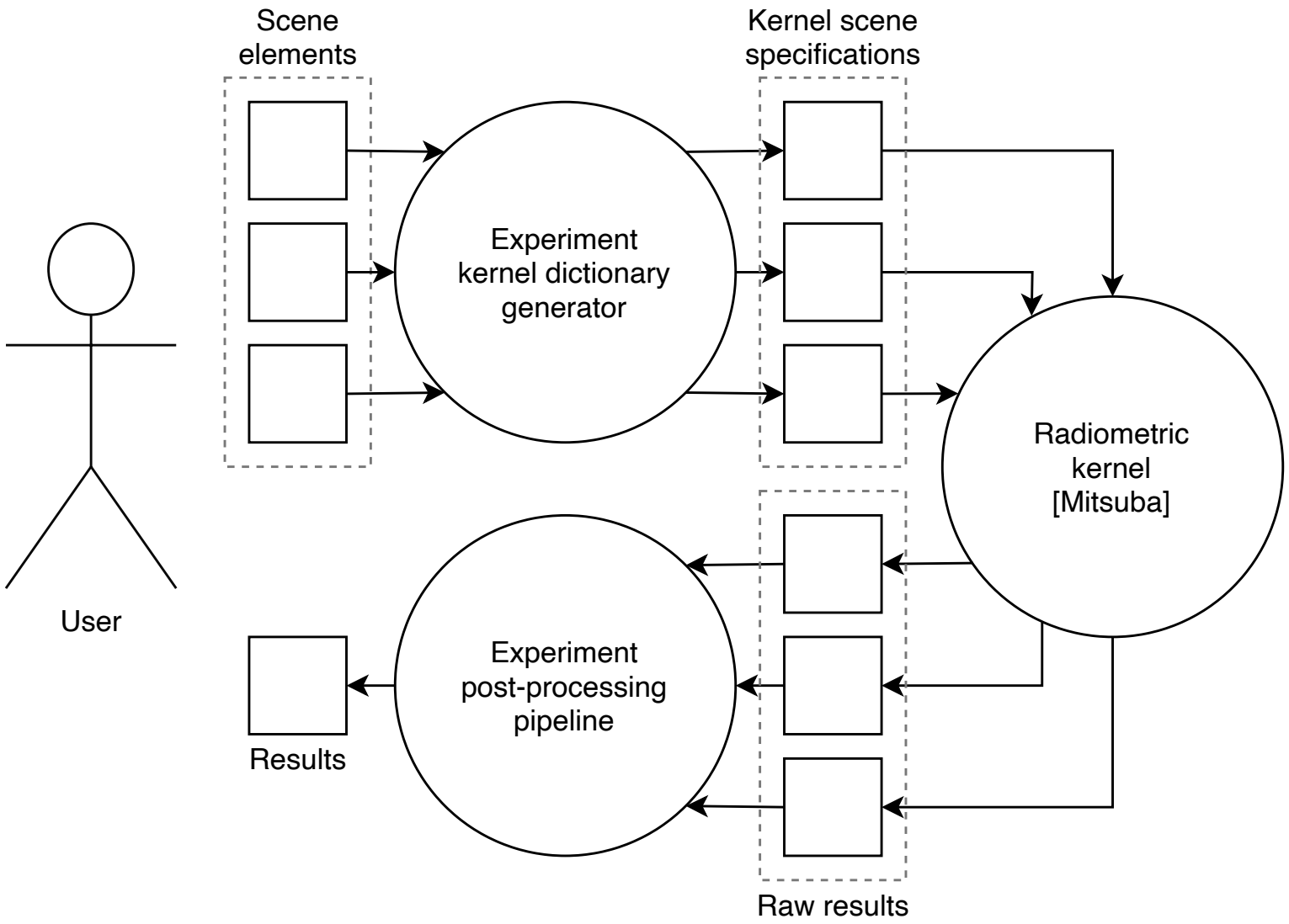
Basic concepts [II]

- **Experiment**s produce a number of **Scene**s passed to Eradiate's Mitsuba radiometric kernel.
- An **Experiment** manages kernel-level simulations based on its many parameters, runs them and post-processes the results.
- Results are packaged as `xarray.Dataset` objects
  ⇒ Very generic, easy integration in existing Python infrastructure

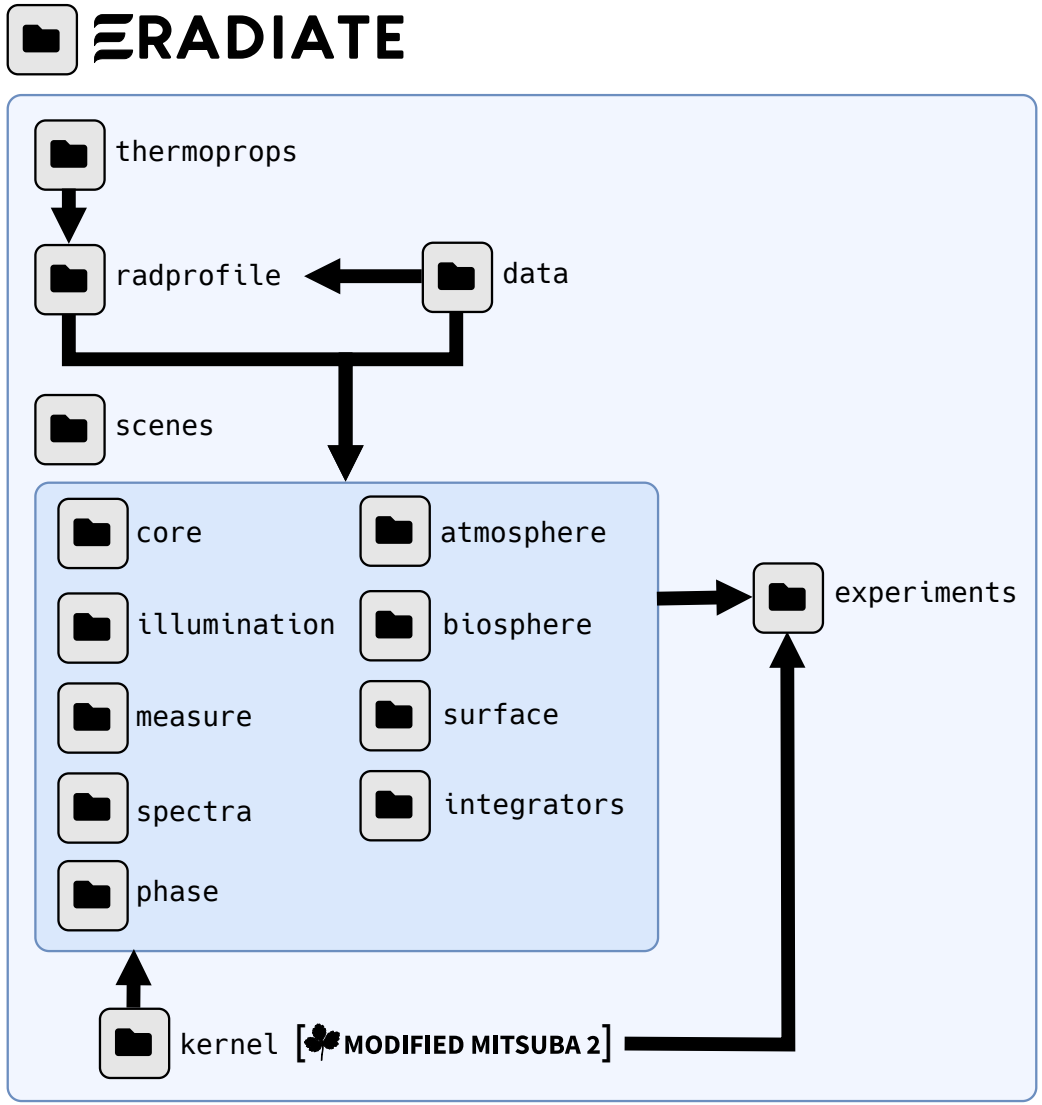
Basic concepts [III]

- **Scene Element**s are used to specify to **Experiment**s how the scene should be set up.
- Specified with higher-level parametrisation than kernel components
  ⇒ More convenient and natural
- Scene elements emit kernel kernel components based on parametrisations relavant to Earth observation, *e.g.*
  - **Measure**: kernel sensors
  - **Illumination**: kernel emitters
  - **Atmosphere**: kernel shapes, medium specification and radiative properties
  - etc.

Data flow overview



Package overview



Documentation

- Please visit https://eradiate.readthedocs.io
- User manual and tutorials currently undergoing updates
  ⇒ many pages are outdated, to be fixed for public release
- API docs are up-to-date ⇒ use *Search* feature
- Documentation of useful dependencies
  - xarray: http://xarray.pydata.org/en/stable/
  - Matplotlib: https://matplotlib.org/

Upcoming development plans for March 2022

- Surface roundedness
  - Because Earth is not flat!
- More data, better data!
  - Support for other AFGL 1986 profile variants
  - More aerosol datasets
  - Improved data management, delivery and customisation
- More tests, better tests!
  - Continuous integration
  - Improved validation
- More docs, better docs!
  - Update tutorials, transfer them to better infrastructure

RAMI4ATM

- Special phase of the RAMI benchmark series
- Goal: Model comparison in conditions closer to satellite observation

For more information, please visit

https://rami-benchmark.jrc.ec.europa.eu/_www/RAMI4ATM.php

FIN

Subscribe to the newsletter for updates
Star/follow us on GitHub
Training material on GitHub