

iceMACS

Collection of tools to calibrate and manage SWIR and VNIR data from the specMACS system, as well as retrieve ice cloud optical properties using a bispectral Nakajima-King retrieval.

Todos

- Change string logs to modern f-string syntax
- Change `open()` calls when reading files to `with open()` in order to ensure files are closed when exception occurs.
- Instead of
- Find a way to avoid `*` imports in `init` file
- Unify LUT generators, preferably into one single function.
- Restructure submodules to avoid confusion. Add classes.
- Add examples
- Complete documentation under usage
- Add a git submodules functionality
- Find better way to organize paths

Usage

SWIR bad pixel interpolation

Many A(C)³ scenes are relatively dark, with a high solar zenith angle and low cirrus radiance values. Some pixels are shown to be unreliable under these conditions. The `PixelInterpolator` class finds these pixels and interpolates for the entire scene. Additionally, interpolation over invalid pixel from the bad pixel list is performed. Initiate with loaded SWIR dataset, containing the variables `radiance` and `valid` access "badness" signal with

```
from iceMACS.tools import PixelInterpolator
interp = PixelInterpolator(swir_ds, window=3)
interp.show_signals()
```

The `window` variable sets the moving average frame size. Choose a fitting cutoff value for each plotted wavelength and pass as ndarray, e.g.

```
interp.add_cutoffs([4, 1.2])
```

Adjust cutoff as needed and apply filter with

```
filtered_radiance = interp.get_filtered_radiance(with_bpl=True)
```

where also interpolating pixels from bad pixel list is default.

Bispectral retrieval (BSR)

Habit detection

Additional functionalities