SPEXone Level 1A to 1C Processor Release Notes

Raul Laasner, Jeroen Rietjens, Otto Hasekamp, Martijn Smit, Richard van Hees, Jochen Landgraf

Citation: Laasner, R., Aan de Brugh J.M.J., Rietjens J., Hasekamp O.P., Smit, M., Van Hees, R.M., Landgraf, J.: SPEXone Level 1A to 1C Processor Release Notes, SRON Netherlands Institute For Space Research, Utrecht, The Netherlands, SRON-SPEXoneL1-2022-02, issue: 1.0, 2022

document number	:	SRON-SPEXoneL1-2022-10-06
software version	:	2.1
date	:	2023-02-10
status	:	final

Contents

1 Introduction

The SPEXone data processor consists of three parts: the L1A-L1B processor, the L1B-L1C processor, and a calibration key data (CKD) generator. The CKD generator uses dedicated on-ground calibration measurements to acquire the CKD. The L1A-L1B processor uses flight L1A data together with the CKD to generate L1B data. The L1B-L1C data collocates L1B data on a common geolocation grid for a predefined reference height.
The L1A-L1B processor and the CKD generator are algorithmically interlinked to ensure full consistency of the derived CKD and the calibration of the L1A data. Therefore, it has been decided to include both elements in the same software package even though the CKD generator is not required for the processing of the SPEXone flight data.

2 New features

2.1 Level 1C

This delivery presents an updated version of the L1B-L1C module of the processor. The new algorithm makes use of a grid cell finding algorithm and a 2D cubic spline to interpolate L1B data onto the L1C target grid.

3 Installation and running

3.1 Dependencies

- GNU C++ compiler version 11
- CMake version 3.13.5
- C netcdf library version 4.7.4 (`netcdf_c++4` and `netcdf`)
- C hdf libraries version 1.10.6 (`hdf5_hl` and `hdf5`)
- Linear algebra library (e.g. lapack but preferably Intel MKL)
- Intel FFT library (optional) — if not available a fallback library is used

3.2 Building

Make a copy of the initial CMake cache file

```
cp initial_cache.cmake.example initial_cache.cmake
```

found in the root source directory and edit it to reflect your environment. Next, create a build directory for configuring and building the SPEXone executable. From the root directory the procedure could look like this:

```
mkdir build && cd build
cmake -C ../initial_cache.cmake -D CMAKE_BUILD_TYPE=release ..
make -j
```

3.3 Running

Start by calibrating one of the L1A products generated by GSFC:

```
mpirun -np 8 <spexone> l1b.yaml
```

where `<spexone>` is the SPEXone processor executable. A GSFC generated L1A product is supplied with delivery but you should be able to use any that has the correct format. The output is a level 1B product that serves as input to the L1B-L1C processor:

```
mpirun -np 1 <spexone> l1c.yaml
```

You can run it with more cores but it doesn't currently scale very well and the calculation is so fast that in practice it might be easier to run several L1B-L1C instances in parallel than trying to push maximum parallel efficiency out of individual runs. In `l1c.yaml` the parameter `l1b_in` refers to the input L1B product. You can use either the reference one or the one you generated yourself. Similarly, using the `l1c_in` keyword, you can switch out the L1C grid file with any other that overlaps with the current orbit.
In order to visualize the geolocation of L1B and L1C products, run

```
python plot_geolocation.py
```

It's a simple plotting script with user adjustable parameters (L1B and L1C products and a viewing angle) found at the top (see Fig. 1 for what to expect). In order to visualize some of the observation data stored in the products, run

```
python plot_I_DoLP.py
```

Figure 2 presents the L1B and L1C radiance and DoLP for a selected wavelength and – in the case of L1C – viewing angle. The L1B quantities are shown for all 8851 along track positions and all viewports. The viewports are stacked horizontally in Fig. 2 in the order -50, -20, 0, 20, 50 deg from left to right. The total number of across track positions is 203 with about 40 per viewport. The scene is mostly uniform with more variation seen for the DoLP, in particular at the swath edges. The L1C scene has 395 ALT positions and corresponds to the L1B ALT range 6305. . . 7209 as visualized in Fig. 1. The white areas of L1C radiance and DoLP are fill values and correspond to L1C locations that are outside the L1B swath. The number of L1C columns outside the range is on average 3 on the left and 4 on the right side of the L1B swath leaving a total of 18 columns with useful data.

4 Files

- `spexone_cal.tar.gz` — source code. Unpack and compile according to instructions in Sec. 3.2.
- `PACE_SPEXONE.20220321T121619.L1A.nc` — L1A product produced at GSFC. This serves as the main input to simulations of this delivery.
- `l1b.yaml` — configuration file for running the L1A-L1B processor.
- `l1c.yaml` — configuration file for running the L1B-L1C processor.

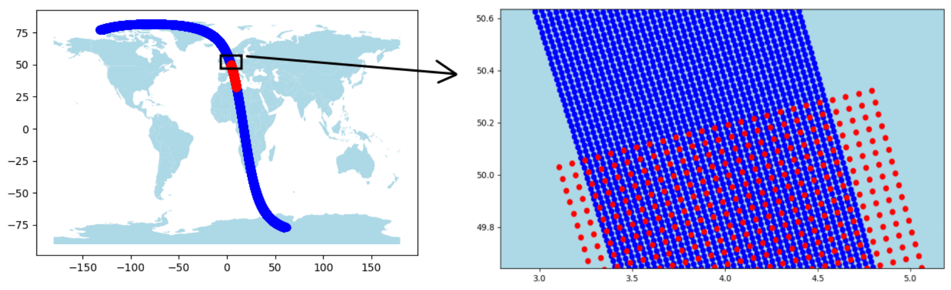

Figure 1: Geolocation (latitudes and longitudes) extracted from L1B (blue) and L1C (red) products.

- `l1b_ref.nc` — reference L1B product. When you run the processor with `l1b.yaml` the output should be identical to this.
- `l1c_ref.nc` — reference L1C product. When you run the processor with `l1c.yaml` the output should be identical to this.
- `ckd.nc` — calibration key data required by the processor when using `l1b.yaml`.
- `PACE_SPEXone.20220321T120000Z.L1C.5.2km.nc` — L1C grid file used as one of the inputs when running the processor with `l1c.yaml`
- `gebco_ocssw_v2020.nc` — elevation map required for geolocation.
- `SPX1_CKD_BIN_TBL_20210304T124000_001.nc` — binning table.
- `UTC.nc` — Map of UTC time difference, constructed out of utcpole.dat from NASA.

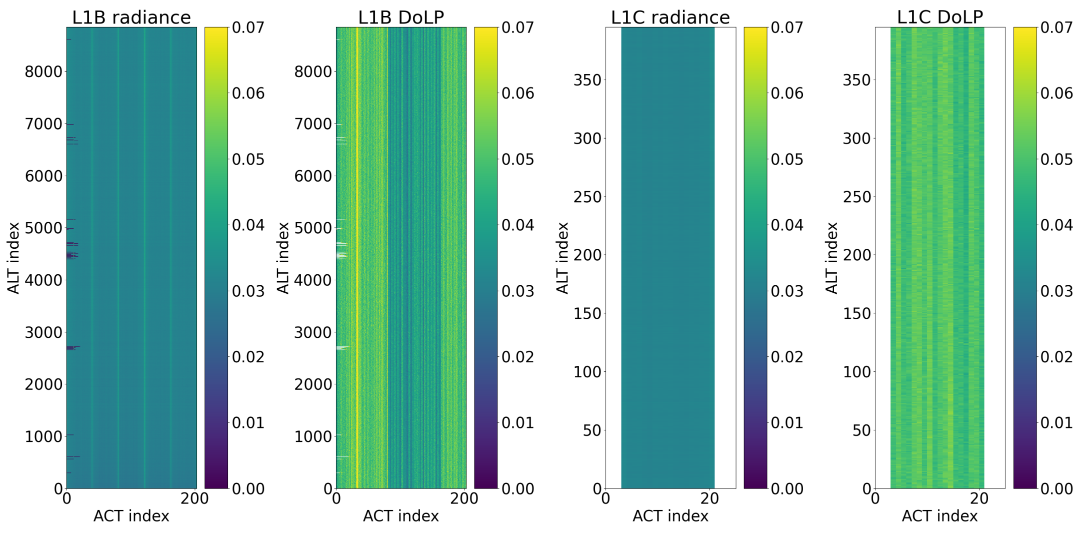

Figure 2: L1B and L1C radiance and DoLP for selected wavelength and viewing angle.