OpenHSI Technical Summary

Christopher Betters
Sydney Photonics Pty Ltd

May 16, 2022

1 Background[1]

OpenHSI is an open-source initiative to increase the accessibility of hyperspectral technology. It consists of a lightweight pushbroom imaging spectrometer using commercial-off-the-shelf components and supporting software to calibrate, capture, and process hyperspectral datacubes on development compute platforms. The project was originally developed by staff and students at the University of Sydney.

The software library *openhsi* provides Remote Sensing practitioners the ability to produce reflectance data using open-source tools such as NASAs 6S radiative transfer model through the Py6S Python wrapper, as well as empirical line calibration. There are also tools to interactively explore datasets and edit metadata.

The University of Sydneys Associate Professor Sergio Leon-Saval and Dr Christopher Betters of the Sydney Astrophotonic Instrumentation Laboratory (SAIL) developed the first OpenHSI camera from previously published designs, adapting it to readily available materials, and thus incrementally changing the design.

[1] Excerpt from `https://openhsi.github.io`

The Australian Research Council funded Training Centre for CubeSats, UAV and their Applications (CUAVA) students Yiwei Mao (PhD Candidate) and Samuel Garske (PhD candidate) developed their software to support the camera as part of their PhD and have publications pending on their work. Professors Iver Cairns and Associate Professor K.C.Wong and Dr Bradley Evans supervise Yiwei and Sam.

2 System Specification

2.1 Summary Table

Parameter	Design Goal^{ab}	Notes
Input Aperture	4mm	
Field Lens Focal Length	16mm	
Sensor Size	1440 x 1080 px, 1.6 MP	Only a 1080 pixel square area is required.
Pixel Size	3.45µm	
FOV	10.7°	
iFOV (along and across-track)	approx. 2 milli-radians (0.1°)	along-track, limited by slit width.
Spatial Samples	>800	slit image length divided by pixel size, binning and image quality my reduce effective samples.
Spectrograph Slit Size (Physical)	3mm by 25µm	
Slit Image Size on sensor	3.1mm by 25.9µm	M=1.035
Spectral Sampling Interval	0.45 nm	nm per pixel
Band Size	3.3 nm	7.5 pixel across slit, corresponds to slit width, ie pixel that need to be binned
Wavelength Range	430 nm to 900 nm	spatial resolution will be degraded blue of 500nm, and beyond 850nm.
Number of Bands	approx 144	
Typical Exposure Time.	10ms	
Signal to Noise estimate	$>$ 150 (430nm to 660nm) $>$ 90 (680nm to 800nm)	10ms exposure, estimated using 6SV solar illumination, Mid latitude Summer, nadir pointing and including detector QE
Size (L x W x H)	135mm x 52mm x 35mm without enclosure	enclosing rectangle; see diagram below for with enclosure
Weight	$<$ 200g without enclosure approx 500g with BR enclosure	
Camera Sensor Model	Phoenix 1.6 MP Model	`https://thinklucid.com/product/phoenix-16-mp-imx273/`

Parameter	Design Goal[ab]	Notes
Digital Interface	1000BASE-T RJ45, PoE ix Industrial, PoE	
Power Requirement	PoE (IEEE 802.3af), or 12-24 VDC external	external via GPIO port, cable not provided.
Power Consumption	3.1W via PoE, 2.5W when powered externally	

3 Getting Started

The Robonation OpenHSI is intended to be used with the OpenHSI software library (https://github.com/openhsi/openhsi). The software library provide methods to acquire and and calibrate raw data from the sensor from digital numbers to radiance (using provided calibration files) or reflectance (using the 6SV code and information the user must provide about atmosphere specify to an observation time and location etc or using an empirical reference for correction). The Github repository has instructions to install and use the library. The team welcomes feedback and bug reports through the Github page.

The Robonation OpenHSI uses a GiGE vision camera manufactured by Lucid Vision Labs, in order to use the OpenHSI package you will also need to acquire the SDK from `https://thinklucid.com/downloads-hub/`.

Once you have the software installed and can acquire data, you can start to acquire images. To do this you must scan your target. This can be achieved by rotating the sensor during an acquisition, sliding an object through the FOV, or by moving the sensor over the ground, for instance on a drone.

3.1 Calibration and Settings files

You will have been provided with a link to default setting and calibration files of your sensor (a pair of json and pkl file for different operating modes).

In the JSON settings file, some of these values should be left as is, as they are part of the calibration (these include: `row_slice`, `resolution`, `win_offset`, `win_resolution`). Other value are configurable: `fwhm_nm`,`exposure_ms`,`luminance`. For the settings `binxy` and `pixel_format`

3.1.1 Modes

Mono8_bin1 Readout bitdepth set to 8 bits and no pixel binning — default mode, but other modes may or may not be more useful.

Mono12_bin1 Readout bitdepth set to 12 bits and no pixel binning — reduces frame rate, but will have better dynamic range.

Mono8_bin2 Readout bitdepth set to 8 bits and 2x2 pixel binning. — increase sensitivity per frame, and increase max frame-rate, but at reduced spatial resolution.

Mono12_bin2 As above, but with bitdepth set to 12 bits. — Camera readout limitation mean not frame rate advantage or disadvantage. You get the maximum possible.

For all of the above there is default JSON file. This is paired with a matching pkl file which contains wavelength and radiance calibration data. There are two version of the pkl file, one for the sensor in the blue robotics housing (appended with `_window`) and another for the sensor without the housing.

4 Linescan Hyperspecteal Imaging: How Does OpenHSI operate.

A linescan camera, like the OpenHSI, records 1 spatial dimension and 1 spectral dimension per frame/line[2]. This is illustrated in Fig. 1, where the field lens project an image of the ground onto a slit. The sensor then records a spectral dispersed image of the slit.

In order to build a full 2D spatial image, the sensor (or the object) must scanned along the slit, hence the nomenclature line-scan sensor/imager/camera[3].

Figure 2 shows sample RGB image generated from a hyperspectral datacube. The bottom part of the figure shows a single frame/line, and regions binned/summed to make the RGB image.

Some common definitions you might see in hyper/multi-spectral literature etc.

Across-track direction This is the dimension of the sensor orthogonal to the direction of scanning, and thus the along the length of the slit. It is the spatial dimension recorded in a single frame/line.

Along-Track direction The is the spatial dimension generated by taking multiple frames and line and parallels to direction of scanning.

FOV The field of view of the sensor for single frame/line, generally in the across track direction. This controlled by the length of the slit and focal length of the field lens.

iFOV The instantaneous field of view of the sensor, in the along-track direction. This corresponds to the width of the slit, and limits the effective spatial resolution of the sensor.

datacube The full 3D (2 spatial, 1 spectral) dataset that is generated when taking a hyperspectral image.

[2] a single frame is sometime referred to as a **line** in the literature

[3] Pushbroom, comes from the action of sweeping a broom, which is analogous to the scan the sensor must preform to cover an area

[4] Modified from Diagram by Lucasbosch, CC BY-SA 4.0 `https://creativecommons.org/licenses/by-sa/4.0`, via Wikimedia Commons, `https://commons.wikimedia.org/wiki/File:Multispectral_imaging_approaches.svg`

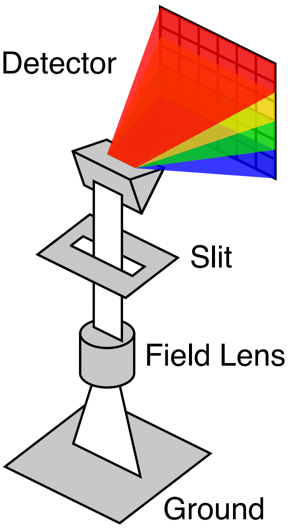

Figure 1: Schematic showing the capture of single frame/line using a line-scan/push-broom hyper-spectral imager[4]. Note, an image is only produced by taking multiple successive frame/lines.

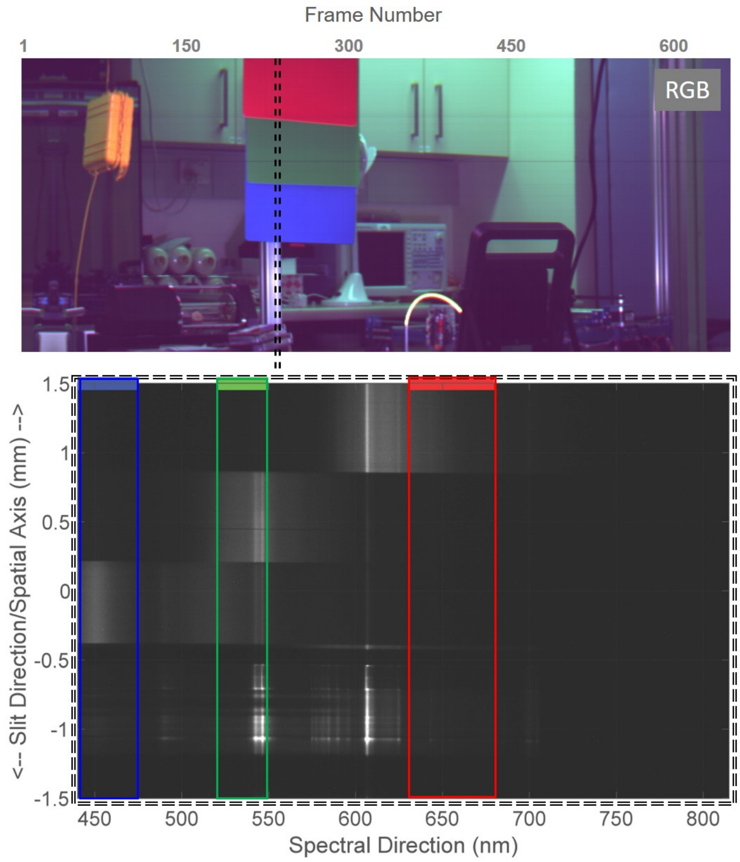

Figure 2: *(top)* RGB image generated from a scan taken with the OpenHSI in our lab.
(bottom) Single frame taken from a scan taken with the OpenHSI in our lab. Frame corresponds to column 250, through the red, green and blue cards. The spectral direction is clipped to 440nm to 815nm

5 WTE Enclosure

The OpenHSI can be removed from the supplied water tight enclosure (WTE), but some care should be taken when removing the back conver or the ethernet connector can be damaged.

Step 1 Remove pressure release valve.

Step 2 Remove the locking cord.

Step 3 Hold the tube and back cover firmly and gently pull while twisting back and forth. The back cover should slowly work its ways off, with out a sudden release. You want to be careful to not pull to hard as a resulting a sudden release which can damage the ethernet connector.

Step 4 Use flat object like a small spatula or flat head screw driver to release the ethernet connect while gently pulling. If you depress the tap and connector should pop out easily.

A video of the process can be found here: `https://bit.ly/3lbXUx0`.