Derya Akkaynak + Stephanie Palmer
Updated: August 2017

Hanlon Lab HSI - Calibration Document

The goal of this document is to have the most up-to-date information in one place to prevent further frustration and delays for anyone wanting to use the Hanlon Lab HSI to do exciting science.

General Notes

Where is the code?

The code that performs the calibration operations for the Hanlon Lab HSI, and the required calibration files live in this gitHub repository:

How do I run the code?

To use this code, download the entire contents of this repository and place them on your local disk. Be sure to keep all the files and folders together. Open the script named ***hanlonHSI_main.m***. This script is ready to be used with the folder of images you would like to process. All you need to do is update the path for the variable named “folder”, pointing to the location of your .3d images.

Where are the outputs saved?

Processed images (datacubes), and a representative RGB image for each datacube is saved in the same directory s above; “folder”. This can be changed by modifying “saveFolder”; simply type the path of the new folder you would like outputs saved in.

Can I modify the code?

You may modify the code as you wish in your local downloaded copy. When you have tested all your modifications and make sure the produced results make sense, you may upload your version to the gitHub repository above, making a new branch. Please generously comment your code, and also provide detailed information in the gitHub versions you change. This is crucial; please respect other people’s time and document all your changes in a human readable way.

What do I do when calibration files change?

First, ensure that the new calibration file is uploaded to the gitHub repository, into the folder named **Hanlon Lab HSI calibration files** since we want to keep all relevant files in a central place. That way, all users have access to it.

Second, make sure the file is named according to the following convention:

hanlonHSI_hanlonHSI_radianceCalibrationFile.mat
hanlonHSI_flatfieldCalibrationFile.mat

And so on. If there is more than one version of a file, use a meaningful suffix. For example:

hanlonHSI_hanlonHSI_radianceCalibrationFile_2017.mat

Third, simply change the pointer to the calibration file whose path or name has changed in the main script ***hanlonHSI_main.m.***

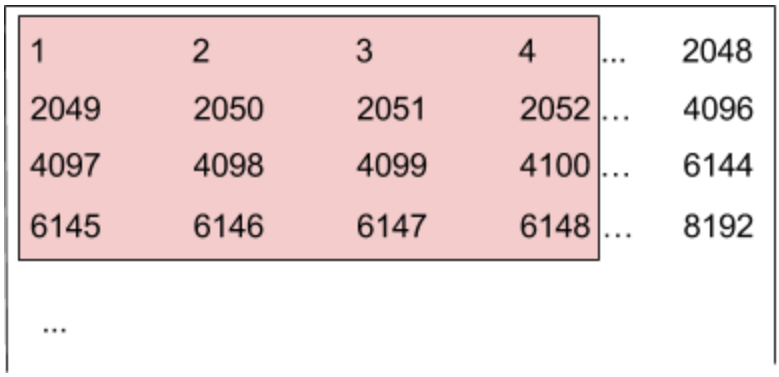
Sensor pattern arrangement

The raw (mosaicked) image we get from the camera is of size 2048 x 2048. This has to be demosaicked into layers of the size 512 x 512 x 16. Surface Optics provided us the order of bands as follows:

560	620	660	436
520	405	380	640
540	360	460	580
600	420	480	500

Descrambling the raw image (***hanlonHSI_parseDataCube.m***)

The raw (mosaicked) image of the size 2048 x 2048 from the camera is in this layout when read into Matlab, where the red rectangle marks one spatial pixel. What is shown is the linear indices:



Thus, the descrambling of the indices of the sensor pattern provided by Surface Optics should be as follows:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Then, the descrambling vector corresponding to wavelengths

[360 380 405 420 436 460 480 500 520 540 560 580 600 620 640 660]

is:

[10 7 6 14 4 11 15 16 5 9 1 12 13 2 8 3].

Spatial corrections

Dark noise correction (***hanlonHSI_darkNoiseCalibration.m***)

Dark noise correction removes the sensor noise that becomes a part of every photo. This operation is integration time (IT) dependent as the sensor noise is amplified with longer IT. The calibration files contain a dark image at every possible IT (recorded with the lens cap on the camera on, ensuring no light leaks in from anywhere else). Then when a novel image is taken, the dark frame image corresponding to the IT it was taken with is subtracted.

Flatfield correction (***hanIonHSI_flatfieldCalibration.m***)

Flatfield correction removes vignetting effects we get through the lens. The flatfield calibration file is a raw image of a gray target that was provided to us by Surface Optics. This image is of size 2048 x 2048. We unscramble it the same way as the raw datacube. Once descrambled, each spectral band is multiplied pixelwise with the flatfield calibration image.

Bad pixel correction (***hanIonHSI_badPixelCalibration.m***)

There are some dead pixels on our sensor, which will always create spatial gaps in our images. The bad pixel calibration file is a hand-made binary mask showing where bad (dead) pixels are. Good pixels are marked with 1, and dead pixels are marked with NaN. Each spectral band of the datacube is multiplied by the bad pixel mask, setting the dead pixels to NaN.

Spectral corrections

Deconvolution (***hanIonHSI_deconvolution.m***)

Deconvolution correction removes artifacts from cross-talk across spectral bands. The calibration file for deconvolution is a 16 x 16 transformation matrix provided to us by Surface Optics. The deconvolution operation is carried out as:
dataCube = dataCube * transpose(deconvolutionMatrix);

Radiance correction (***hanIonHSI_radianceConversion.m***)

The filters that split the light into 16 wavelengths do not keep the amplitude of the incoming signal equal for each wavelength. Thus, the intensity is lower for some bands than for others. The radiance calibration file is provided to us from Surface Optics scales the resulting image for this. As of August 6, 2017, the radiance calibration file we have from Surface Optics does not work for images taken beyond 2015, and we are awaiting a new calibration file to process images taken since then.

sRGB image synthesis (***hanIonHSI_synthesizeRGBImg.m***)

Once the calibration steps mentioned above are done, we can synthesize an RGB image to give us an idea of the scene. The linear RGB image is generated by combining the bands corresponding to 460, 540, and 640 nm. Then this linear RGB image is companded into sRGB color space. This makes it appear more crisp, and more aligned with the kind of images we are used to seeing. Also, since the RGB image is for visualization purposes and not used as data, the bad pixels are smoothed out using median filtering. This can be turned off if needed.

Saving outputs (***hanIonHSI_saveCalibratedImage.m***)

The following outputs are saved; calibrated data cube (.mat file), rgb image (.mat file), rgb image (.jpg file) in the folder “saveFolder”. By default, this is the same as the folder the user specified for .3d images. It may be changed in the **hanIonHSI_main.m** script as needed.