High-level simulation of future multispectral imagery from the proposed TreeView satellite.

Joe Fennell

December 16, 2020

Abstract

This document describes the early work carried out for the development of a simulation-evaluation pipeline for earth orbit multispectral imagery. A description of the pipeline is presented and example outputs.

1 Introduction

The TreeView project proposes a novel satellite solution for climate action that will drive a revolution in 'Precision Forestry' –- the use of advanced technologies for a more granular data capture and management. Forests are often considered large, remote areas that require low maintenance during their lifetime and can be managed in homogeneous units called forest stands. But new political, environmental, social and commercial drivers necessitate a more diverse approach of mixed species and smaller scale planting. Individual trees are the fundamental unit of these resources, and the target of TreeView.

1.1 Work Package 2000: Science Requirements

Current work (Q4 2020) focuses on the need to constrain the engineering design with understanding of the likely measurements that will be made with the proposed system. In order to do this, both a simulation methodology and a set of relevant performance metrics must be adopted, such that different combinations of band-pass filter and sensor characteristics can be evaluated for different geographically-relevant atmospheric conditions.

In designing the work the following considerations were made:

1. As this project relates specifically to the measurement of single trees (as opposed to forest stands), tree stands and individual isolated trees of various sizes will both be considered when designing performance metric experiments
2. Due to the limited knowledge of the likely system architectures at this stage in the development process, a high-level approach will be taken to simulation of synthetic images with assumptions made about the likely distributions and properties of noise at different stages in the simulation.
3. The range of possible spectral sensitivities will be limited to what is feasible with a silicon detector, and observable through standard glass optics ($\lambda \approx 400 - 900$ nm).
4. The simulation pipeline is designed using Dask/Python as it is (i) Open Source (and so easily verified, reviewed and tested by any research group) and (ii) a natively parallel environment for efficient, scalable computation.

2 Stages of Simulation

This section details the tools and methodologies used for each stage in the simulation/evaluation pipeline.

2.1 Imagery Collection

Hyperspectral aerial imagery from the project partners 2ExcelGeo have been used as the main ref-

erence data for these simulations. The relevant datasets from the aerial collect were the HySpex VNIR covering a spectral range covering the 400-900 nm band and at a ground sampling distance of $0.33 mpixel^{-1}$ - much higher spatial and spectral resolution than the proposed mission. Imagery have been collected over a number of UK cities including Milton Keynes and Birmingham. Time sequences (multiple images of the same ground area) have been collected over some areas and will be made available to this team within the current project.

The measured at-sensor radiance is converted to a reflectance value through an atmospheric correction model implemented in ENVI software (QUAC [ref needed]). This is carried out by the data providers (2ExcelGeo). The QUAC method is based on the use of lookup tables of pure endmember reflectances and their corresponding radiance under known solar conditions. The ill-posed problem is then treated as a spectral-unmixing of known endmembers. It is favoured for it's low computation demand, however there is some concern that the approach may introduce spectral artefacts to the imagery that are hard to quantify and studies have shown worse performance than conventional radiative transfer model inversion methods such as FLAASH [citation needed]. More work needed to understand if a more rigorous atmospheric approach is needed for these data.

2.2 Atmospheric Simulation

2.2.1 Reflectance to at-sensor radiance

In broad terms, the purpose of the atmospheric simulation is to map from ground reflectance (ρ) to an at-sensor radiance ($L_{e,\Omega}$):

$$\rho(\lambda) \rightarrow L_{e,\Omega}(\lambda) \tag{1}$$

Established radiative transfer modelling approaches were used in the simulation. Primarily, the 6SV [ref needed] code from NASA was used to convert the measured ground reflectance to at-sensor radiance outside the atmosphere. The code is considered a medium-complexity, line-by-line approach and is used for atmospheric correction of MODIS data. For practical implementation reasons, the 6SV code (a FORTRAN library accessed via the Py6S bindings) has been used to generate Lookup Tables (LUTs) for a limited range of parameter values.

In the first stages, the following approximations are proposed:

1. The satellite observes each pixel in space, at nadir. If the centre of the image plane is assumed to be directly beneath the satellite's orbital path, then the inaccuracy introduced by this approximation increases towards the edges of the image plane. This is a reasonable approximation for a narrow field-of-view, high resolution device as proposed.

2. The atmospheric optical thickness will be set at a range typical of the UK. All test data is of UK locations and the UK's climatic conditions are quite challenging for optical imaging. This is therefore a useful starting point for simulation.

3. All individual pixels in the imagery array will be considered as independent homogeneous lambertian diffusers for the purpose of the 6SV simulation. As such, the spatial response function is handled at a later stage in the pipeline.

4. The simulation is evaluated for all wavelengths between

2.3 Satellite Geometric Simulation

The at-satellite radiance ($L_{e,\Omega}(\lambda)$) is in units of power per steradian (Wsr^{-1}) and needs to be converted to at-satellite irradiance (Wm^{-2}). Each cell in the input array (ground plane) corresponds to a fraction of the sensor field of view (Instantaneous field-of-view – IFoV) (sr). As such, the per-cell irradiance (E_i) should be estimated for a sensor at nadir using the Ground Sample Distance (d_{GSD}) and the altitude of the satellite above the target(d_A):

$$E_e(\lambda) = L_{e,\Omega}(\lambda) \tan^{-1}\left(\frac{d_{GSD}}{d_A}\right) \tag{2}$$

As the ground plane represents a non-zero area, this is an approximation for pixels not at the centre of the image.

2.4 Sensor Simulation

The stages of the sensor simulation are given below. Only linear responses (with the exception of physically-motivated truncations) have been modelled here to represent the linear response component of the sensor. Fixed pattern noise has not been modelled.

2.4.1 Sensor radiance to photon flux

Initially, pixels in the input array are treated independently and the following stages are carried out element-wise with every pixel treated as a 1D spectrum. The at-sensor irradiance is converted to at-sensor photon rate ($\Phi_0(\lambda)$) using the Planck-Einstein relation:

$$\Phi_0(\lambda) = E_e(\lambda)\frac{\lambda}{hc} \tag{3}$$

2.4.2 Application of bandpass response functions

For every band, the bandpass transmittance is multiplied element-wise by the flux and integrated w.r.t. wavelength

$$\Phi_{band} = \int_{400}^{900} (\Phi_0(\lambda)T_i(\lambda))\ d\lambda \tag{4}$$

where T_i is the transmission spectrum of the ith bandpass filter. This provides the per-band photon flux.

2.4.3 Spatial Resampling

The spatial response function takes into account atmospheric blurring, optical imperfections, motion blur, etc. of the entire system.

To convert from the ground plane to image plane, the integrated fluxes for each band (Φ_{band}) are separately convolved with the spatial response function, $g(x, y)$.

$$\Phi_{band}(x, y) = g(x, y) * \Phi_{band}(x, y) \tag{5}$$

where x, y are ground position indices.

In the absence of more information about the optical properties, the spatial response function ($g(x, y)$) has been approximated as a 2D gaussian with parameters derived empirically. An example of this using the Sentinel 2 Point Spread Function estimated the 10m ground sampling distance Sentinel 2 bands to have a Full Width at Half Maximum (FWHM) of 22.08 m and this ratio was used for preliminary estimates of different ground sampling distances.

The FWHM can be converted to sigma:

$$\sigma_{xy} = \frac{(FWHM/2.355)}{dx} \tag{6}$$

where $dx = dy$
and used to generate the gaussian kernel:

$$g(x) = \exp\left(-\left(\frac{(x - x_o)^2}{2\sigma_{xy}^2} + \frac{(y - y_o)^2}{2\sigma_{xy}^2}\right)\right) \tag{7}$$

After convolution, the new array is then resampled by taking the local mean over a subset of cells representing the area of one sensor pixel.

2.4.4 Pixel photon count

To convert from pixel flux to pixel photon count:

$$\bar{I_0} = \Phi_{band} t A \tag{8}$$

where t is integration time and A is the area of the pixel. The quantity $\bar{I_0}$ describes the mean number of photons hitting a pixel in any given integration cycle.

2.4.5 Photon noise

The photon count at the sensor can be expressed as a Poisson process such that the Λ coefficient of each process is the observed count for a given pixel

$$I_0(x, y, band) \sim Poisson(\Lambda) \tag{9}$$

Where the estimated photon quantity at each pixel, $I_0(x, y, band)$ is the Λ coefficient for each distribution.

2.4.6 Photon to electron

To convert from photon to electron, given a known sensor quantum efficiency:

$$I_1 = round(I_0 \cdot \bar{Q}_{E,i}) \tag{10}$$

where $\bar{Q}_{E,i}$ is the weighted mean of the quantum efficiency in the ith channel:

$$\bar{Q}_{E,i} = \sum_{\lambda=300}^{900} Q_E(\lambda) \left(\frac{T_i(\lambda)}{\int_{300}^{700} T_i(\lambda) d\lambda} \right) \quad (11)$$

2.4.7 Dark Current Noise

The contribution of dark current electrons is assumed to be a additive gaussian with zero mean and standard deviation:

$$I_{dark} \sim N(0, \sigma) \quad (12)$$

2.4.8 Conversion to Voltage

The signals are added and truncated at the full well capacity of the sensor

$$I_2 = trunc(I_1 + I_{dark}, C) \quad (13)$$

where C is the sensor specific well capacity.

The charge collected during an integration cycle is then converted to a voltage

$$V_s = V_{SN} - I_2 \cdot G_{SN} \quad (14)$$

where V_{SN} is the reference voltage (V) and G_{SN} is the sense node gain (V/e^-).

2.4.9 Conversion to Digital Number

The amplified voltage can then be converted to a digital number (DN).

$$DN = trunc(V_{ADC} - V_s \cdot G_{ADC}, DN_{max}) \quad (15)$$

where V_{ADC} is the Analogue-Digital Converter reference voltage and G_{ADC} is the ADC gain (DN/V).

The result is truncated at the maximum DN value (DN_{max}) and zero to simulate clipping and non-detection behaviours.

2.4.10 Conversion to reflectance

For testing and evaluation purposes, the pipeline is used to generate a sensor response for a 100% reference image (i.e. all atmospheric and sensor parameters are known *a priori*). The image signal is then divided by reference signal to produce the estimate of reflectance at surface level.

$$\rho = \frac{DN}{DN_{100}} \quad (16)$$

Noisy reference spectra can easily be generated by replacing fixed parameters with a distribution from which instances are sampled, for example to explore effect of uncertainties in Atmospheric Optical Thickness (AOT) measurement.

3 Evaluating Performance

Performance metrics for a given sensor-atmosphere parametrisation can be divided into classification and parameter estimation metrics.

3.1 Classification

No development

3.2 Parameter estimation

A spectral index (defined by narrowband wavelength or wavelength ranges) is first estimated from the hyperspectral data and is used as the reference value. An analagous definition for the sensor is then separately defined.

For example for Normalised Difference Vegetation Index (NDVI):

$$S = \frac{\rho_{IR} - \rho_R}{\rho_{IR} + \rho_R} \quad (17)$$

where ρ can be defined differently for each sensor.

As a calibration experiment would normally be carried out to account for the different definitions of ρ between sensors, this can be simulated by fitting a linear regression model between predicted and observed values:

$$S_{sim} = \beta_0 S_{ref} + \beta_1 \quad (18)$$

where β_0 and β_1 are free parameters.

As the models simulate complex noise distributions and the number of observations is high, a robust regression algorithm is adopted (Random Sample Consensus - RanSaC) to exclude outlying values.

Prior to model fitting, the (assumed lower) resolution simulation is resampled to the same spatial resolution as the reference using nearest-neighbour resampling.

The model is then fitted as described in [RANSAC ref Needed] and raw and calibrated Root Mean Squared Errors (RMSEs) are calculated:

$$RMSE = \sqrt{\frac{(S_{ref} - S_{sim})^2}{N}} \tag{19}$$

4 Sample Results

4.1 Image Generation

The pipeline was used to generate test signals for a low noise (10 $e^- pixel^{-1}$, low spatial resolution (4m Ground Sample Distance) scenario (Figure 1) and a high noise (120 $e^- pixel^{-1}$, low spatial resolution (1m Ground Sample Distance) scenario (Figure 2). Radiances for a satellite position perpendicular to the centre of the sample imagery (52.04N, 0.76E) on the day the imagery were collected (22/6/2020) at solar noon (12.00 UST) and with a standard coastal atmospheric optical profile were generated for estimating at-sensor radiances.

4.2 Demonstration 1: Effect of Dark Noise level

A simple experiment is presented here, demonstrating how the pipeline can be used to score different sensor properties. In this experiment, the dark noise level (σ_{dark}) was varied between 0 and 120 electrons per pixel for the scenarios presented in Tables 1 and 3. The scoring metric used was the Root Mean Squared Error (RMSE) of the Normalised Difference Vegetation Index (NDVI). Each parameter combination was run 10 times (Figure 3).

4.2.1 Performance

The model, for a 32,400 pixel input image was run 240 times (2 scenarios, 10 repeats, 12 parameter combinations) taking approximately 52 seconds on a 4-core 2.3GHz i5 macbook. Speed increases are likely to be possible once the expected range of parameters to be evaluated are known. Currently the image simulation stage is optimised for parallel use, however the evaluation pipeline needs to be updated to make fully parallel.

4.2.2 Calibration

Use of RanSaC calibration decreased the RMSE for both scenarios (Figure 3). The lowest RMSE appeared to be at the extreme NDVI values (Figure 4). This is expected as the mid-value NDVI values would be generated where the spatial downsampling includes a mixture of extreme values (e.g. the edge between a building and vegetation). The calibration could be improved by fitting piece-wise regression models to important parameter ranges (e.g. the NDVI values of plants) rather than the whole range.

4.2.3 Effect of Dark Noise

For the zero-noise, the 1m scenario had a calibrated RMSE of 0.12 compared to 0.26 in the 5m scenario (Standard Deviation < 0.001). For increasing noise levels between 11 and 120 e^-, the calibrated RMSE increased in both scenarios, however the increase was greater in the high spatial resolution example, rising to 0.23 at $120e^-$. This was expected as the spatial degradation noise in the lower spatial resolution scenario already dominates. Even a completely noise-free example will have a RMSE > 1 due to spatial downsampling kernel.

5 Conclusions

1. The simulation pipeline generates linear simulations efficiently but needs a full code review. Other potential noise sources that have not been modelled should be considered and included if thought to be significant

2. The evaluation pipeline can be used to test different sensor parameters against spectral index retrieval performance. This will be developed further to score classification performance.

3. More information is needed for the likely ranges of sensor properties.

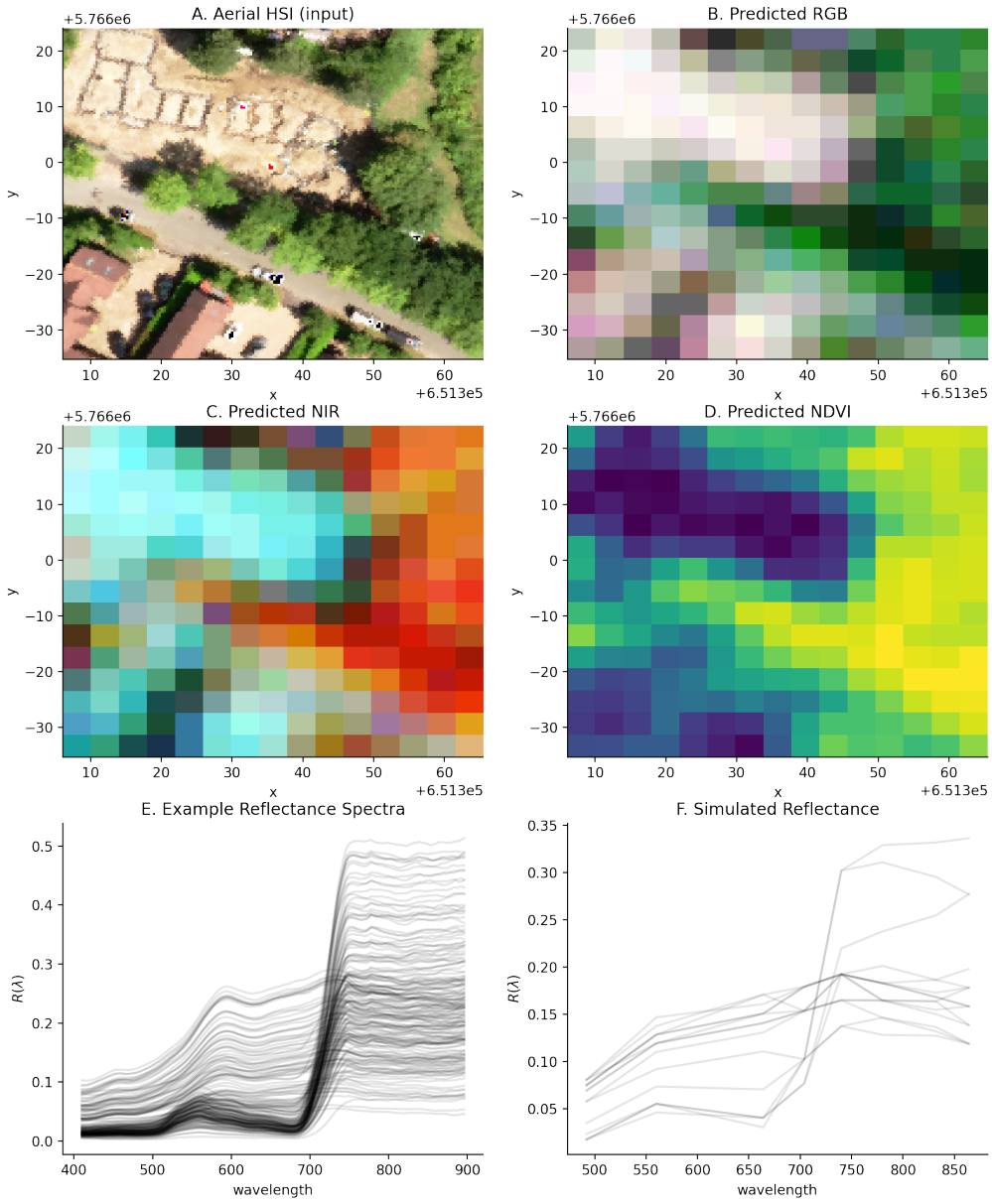

Figure 1: Sample simulated signals (4m GSD, dark noise=10) using a pipeline based on Sentinel-2 VNIR bands and a CCD sensor. The input hyperspectral (A), simulated RGB (B), Predicted false colour NIR (C), Normalised Difference Vegetation Index (D) are shown as images and example input reflectance spectra (E) and example simulated reflectance spectra (F) are shown as line plots.

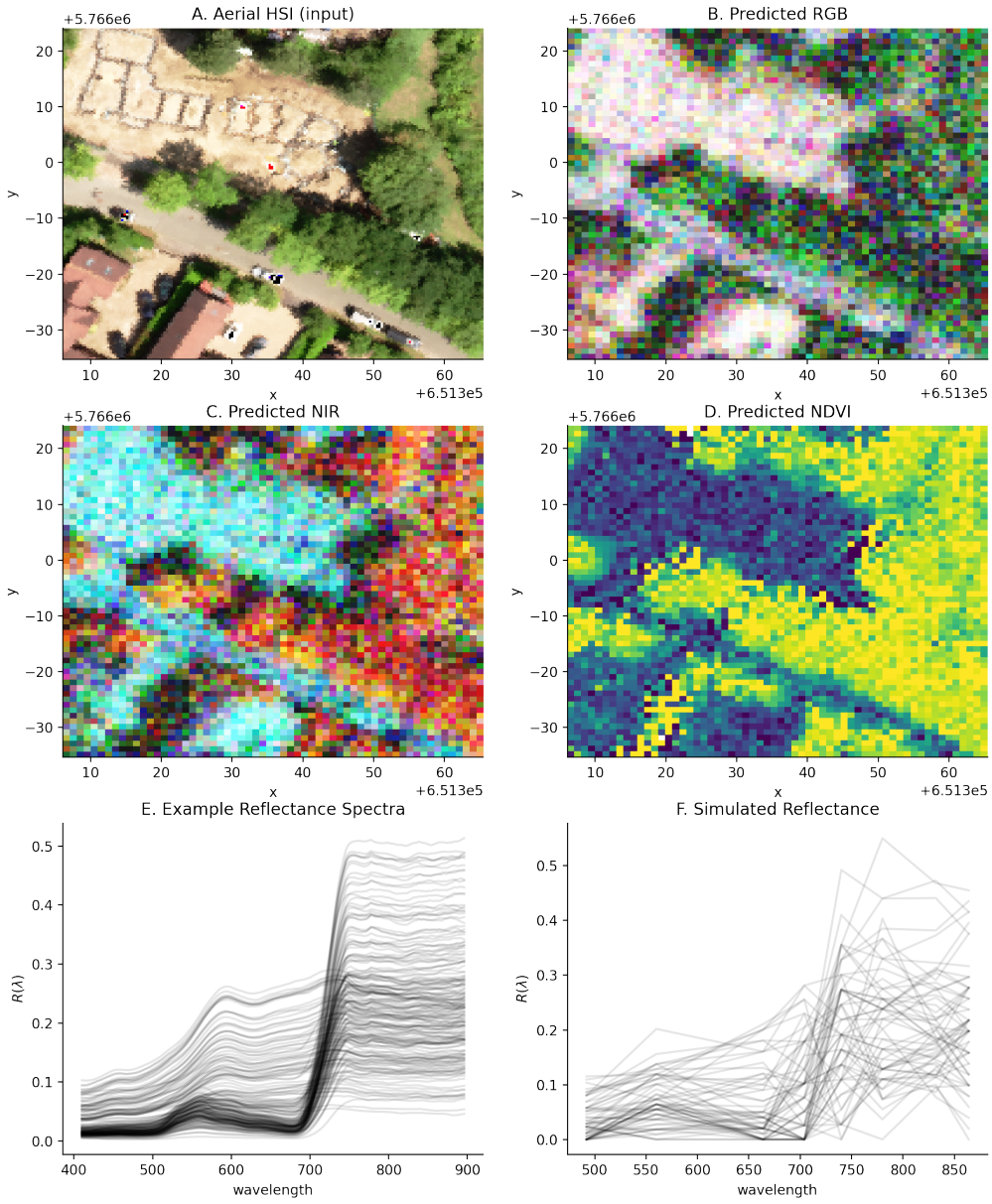

Figure 2: Sample simulated signals (1m GSD, dark noise=120) using a pipeline based on Sentinel-2 VNIR bands and a CCD sensor. The input hyperspectral (A), simulated RGB (B), Predicted false colour NIR (C), Normalised Difference Vegetation Index (D) are shown as images and example input reflectance spectra (E) and example simulated reflectance spectra (F) are shown as line plots.

Table 1: Processing steps (4m GSD, dark noise=10) using a pipeline based on Sentinel-2 VNIR bands and a CCD sensor corresponding to outputs in Figure 1

	Stage	**Function**	**Parameters**
1	irradiance per original pixel	pyeosim.detector.radiance_to_irradiance	altitude = 5.00e+05
2	irradiance to flux	pyeosim.detector.irradiance_to_flux	None
3	flux at CCD	pyeosim.spectral.TreeView_1	None
4	flux at resampled pixel	pyeosim.spatial.gaussian_isotropic	psf_fwhm = 8.00 ground_sample_distance = 4.00
5	flux to quanta	pyeosim.detector.photon_mean	pixel_area = 3.00 integration_time = 0.10
6	photon noise	pyeosim.detector.add_photon_noise	None
7	photon to electron	pyeosim.detector.photon_to_electron	Q_e = [0.87,...]
8	dark current noise	pyeosim.detector.add_gaussian_noise	sigma = 10.00
9	electron to voltage	pyeosim.detector.electron_to_voltage	v_ref = 5.00 sense_node_gain = 5.00e-06 full_well = 1.00e+05
10	voltage to DN	pyeosim.detector.voltage_to_DN	v_ref = 5.00 adc_gain = 5.00e+03 bit_depth = 12.00

Table 2: Processing steps (1m GSD, dark noise=120) using a pipeline based on Sentinel-2 VNIR bands and a CCD sensor corresponding to outputs in Figure 2

	Stage	**Function**	**Parameters**
1	irradiance per original pixel	pyeosim.detector.radiance_to_irradiance	altitude = 5.00e+05
2	irradiance to flux	pyeosim.detector.irradiance_to_flux	None
3	flux at CCD	pyeosim.spectral.TreeView_1	None
4	flux at resampled pixel	pyeosim.spatial.gaussian_isotropic	psf_fwhm = 2.00 ground_sample_distance = 1.00
5	flux to quanta	pyeosim.detector.photon_mean	pixel_area = 3.00 integration_time = 0.10
6	photon noise	pyeosim.detector.add_photon_noise	None
7	photon to electron	pyeosim.detector.photon_to_electron	Q_e = [0.87,...]
8	dark current noise	pyeosim.detector.add_gaussian_noise	sigma = 1.20e+02
9	electron to voltage	pyeosim.detector.electron_to_voltage	v_ref = 5.00 sense_node_gain = 5.00e-06 full_well = 1.00e+05
10	voltage to DN	pyeosim.detector.voltage_to_DN	v_ref = 5.00 adc_gain = 5.00e+03 bit_depth = 12.00

Table 3: Processing steps (1m GSD, dark noise=120) using a pipeline based on Sentinel-2 VNIR bands and a CCD sensor corresponding to outputs in Figure 2

	Stage	**Function**	**Parameters**
1	radiant energy to radiant flux	pyeosim._sensor.energy_to_quantity	None
2	apply bandpass filters	pyeosim.spectral.TreeView_1	None
3	apply spatial resampling	pyeosim.spatial.gaussian_isotropic	psf_fwhm = 4.00 ground_sample_distance = 2.00
4	radiant flux to flux density	pyeosim._sensor.radiance_to_irradiance_2	lens_diameter = 0.10 focal_length = 2.57
5	flux density to flux	pyeosim._sensor.photon_mean	pixel_area = 100.00 integration_time = 9.14e-03
6	add photon shot noise	pyeosim._sensor.add_photon_noise	None
7	photon to electron	pyeosim._sensor.photon_to_electron	Q_E = [0.86, ...]
8	add photo response non-uniformity	pyeosim._sensor.add_prnu	prnu = [-0.00, ...]
9	add dark signal	pyeosim._sensor.add_dark_signal	dark_current = 1.82e+04 integration_time = 9.14e-03 dsnu = [0.00, ...]
10	electron to voltage	pyeosim._sensor.electron_to_voltage_ktc	v_ref = 3.10 sense_node_gain = 5.00e-06 full_well = 3.00e+04 temperature = 2.93e+02
11	add column offset noise	pyeosim._sensor.add_column_offset	offset = [-0.00, ...]
12	voltage to DN	pyeosim._sensor.voltage_to_DN	v_ref = 3.10 adc_gain = 1.00e+05 bit_depth = 12.00

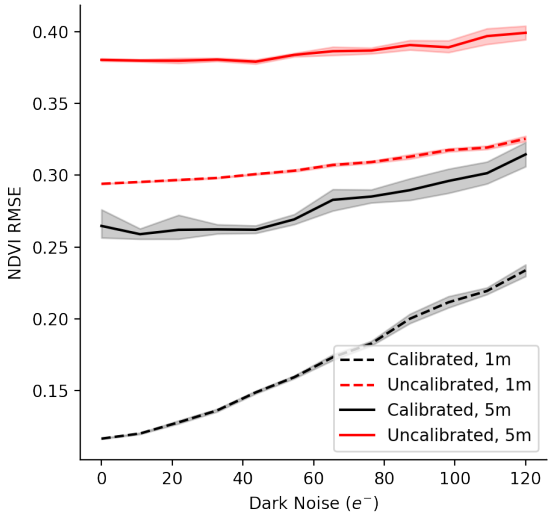

Figure 3: Uncalibrated (red) and calibrated (black) Root Mean Squared Error for low spatial resolution (solid) and high spatial resolution (dashed) scenarios over a range of dark noise levels

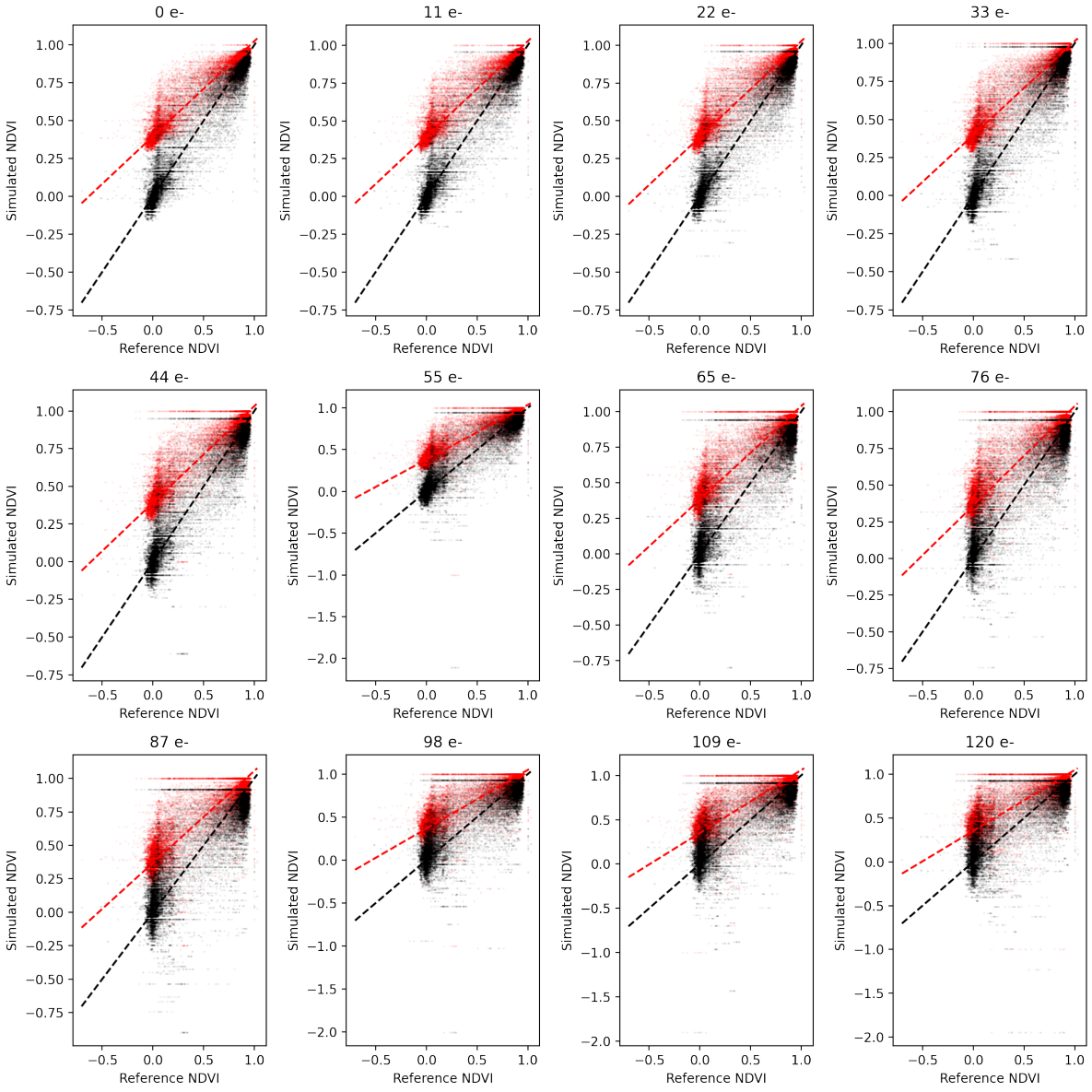

Figure 4: Uncalibrated (red points) and calibrated (black) simulated v. reference values with regression lines showing model fits. Panels show different noise levels