sensors

MDPI

Article

A Customisable Data Acquisition System for Open-Source Hyperspectral Imaging

Yiwei Mao [1,2], Christopher H. Betters [1,2], Samuel Garske [2], Jeremy Randle [3], K. C. Wong [2], Iver H. Cairns [1,2,*] and Bradley J. Evans [2,4]

[1] School of Physics, The University of Sydney, Sydney, NSW 2006, Australia; yiwei.mao@sydney.edu.au (Y.M.); christopher.betters@sydney.edu.au (C.H.B.)
[2] ARC Training Centre for CubeSats UAVs and Their Applications, The University of Sydney, Sydney, NSW 2006, Australia; sam.garske@sydney.edu.au (S.G.); kc.wong@sydney.edu.au (K.C.W.); bradley.evans@une.edu.au (B.J.E.)
[3] Australian Centre for Field Robotics, The University of Sydney, Sydney, NSW 2006, Australia; jag@acfr.usyd.edu.au
[4] School of Environment and Rural Science, University of New England, Armidale, NSW 2351, Australia
* Correspondence: iver.cairns@sydney.edu.au

Abstract: Hyperspectral imagers, or imaging spectrometers, are used in many remote sensing environmental studies in fields such as agriculture, forestry, geology, and hydrology. In recent years, compact hyperspectral imagers were developed using commercial-off-the-shelf components, but there are not yet any off-the-shelf data acquisition systems on the market to deploy them. The lack of a self-contained data acquisition system with navigation sensors is a challenge that needs to be overcome to successfully deploy these sensors on remote platforms such as drones and aircraft. Our work is the first successful attempt to deploy an entirely open-source system that is able to collect hyperspectral and navigation data concurrently for direct georeferencing. In this paper, we describe a low-cost, lightweight, and deployable data acquisition device for the open-source hyperspectral imager (OpenHSI). We utilised commercial-off-the-shelf hardware and open-source software to create a compact data acquisition device that can be easily transported and deployed. The device includes a microcontroller and a custom-designed PCB board to interface with ancillary sensors and a Raspberry Pi 4B/NVIDIA Jetson. We demonstrated our data acquisition system on a Matrice M600 drone at a beach in Sydney, Australia, collecting timestamped hyperspectral, navigation, and orientation data in parallel. Using the navigation and orientation data, the hyperspectral data were georeferenced. While the entire system including the pushbroom hyperspectral imager and housing weighed 735 g, it was designed to be easy to assemble and modify. This low-cost, customisable, deployable data acquisition system provides a cost-effective solution for the remote sensing of hyperspectral data for everyone.

Keywords: data acquisition; hyperspectral imaging; remote sensing; open-source

check for updates

Citation: Mao, Y.; Betters, C.H.; Garske, S.; Randle, J.; Wong, K.C.; Cairns, I.H.; Evans, B.J. A Customisable Data Acquisition System for Open-Source Hyperspectral Imaging. *Sensors* **2023**, *23*, 8622. https://doi.org/10.3390/s23208622

Academic Editors: Yuwei Chen, Hui-liang Shen, Siyuan Cao and Eero Ahokas

Received: 16 September 2023
Revised: 17 October 2023
Accepted: 17 October 2023
Published: 21 October 2023

1. Introduction

Hyperspectral imagers, also known as imaging spectrometers, capture imagery in many narrow spectrally contiguous bands. For remote sensing, this presents two challenges: recording high-sample-rate image data and having the ancillary geospatial information to map them. This ability is well-suited for environmental studies and facilitates measurements of the physical properties of objects without physical contact. Imaging spectroscopy is used in mineral mapping [1–5], geology and soils [6–9], plant ecology and invasive species monitoring [10–14], hydrology [15–20], and the atmospheric modelling of pollutants [21–25]. Many remote sensing applications rely on these instruments deployed on satellites, such as Hyperion onboard the Earth Observing-1 (EO-1) satellite [26–28], and mounted on aircraft such as the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [29–31] and HyMAP [32–34].

Recent developments in hyperspectral imager design have made portable low-cost applications possible for studies on leaf composition [35,36], quality control in production [37], aurora observations [38], and urban monitoring [39]. For these compact hyperspectral imagers, sometimes known as do-it-yourself hyperspectral imagers [40–42], pushbroom designs are the most popular, with a few using a snapshot design [43]. Diffraction gratings are commonly used [44], although some use prisms as the dispersing mechanism [45]. Some of these can send visualisations to a mobile device using WiFi [35,46], but this is short range by design.

These devices need to be tethered in order to be powered and operated [47], limiting the available applications and preventing true remote sensing onboard autonomous platforms. Substituting for a drone, Ref. [48] deployed one of these hyperspectral imagers from a mast on a taut rope. They did not collect navigation data, so any motion artefacts were not corrected for. With the exception of [49], all of these custom hyperspectral imaging systems have not been deployed on an autonomous platform. In a review by Stuart et al. on the state of compact field deployable hyperspectral imaging systems [50], the complexity of georeferencing was highlighted, as this requires navigation and orientation data to be collected and time-synchronised to the hyperspectral imagery. The availability of a customisable data acquisition system that is capable of collecting hyperspectral, navigation, and orientation data is holding back the deployment of non-commercial hyperspectral imagers for remote sensing.

Despite being out of reach for most remote sensing practitioners and requiring a larger budget, it is worth mentioning that hyperspectral imagers built with COTS components were proposed for CubeSats [51–54].

Having a standalone data acquisition system to collect navigation and orientation data has a few advantages over extraction from DJI or ArduPilot data logs. The hyperspectral data and ancillary navigation data need to be timestamped and synchronised for accurate georeferencing [55], since time offsets lead to spatial offsets. Therefore, we used the global navigation satellite system (GNSS) time and pulse per second (PPS) signal to synchronise the onboard computer system time. Secondly, in order to accurately model the camera dynamics, the IMU sensor sample rate needs to be faster than the camera frame rate, and this is difficult to control or requires advanced modifications onboard DJI or Ardupilot drones. A customisable data acquisition system allows flexibility in selecting frame rates and exposure times. Furthermore, we were able to measure the orientation of the hyperspectral imager directly by mounting an IMU on the camera body, rather than estimating the orientation by proxy from the UAV or gimbal orientation. These advantages allow our data acquisition system to be deployed beyond UAVs and be used on aircraft and satellites.

While some commercial hyperspectral imagers come with built-in data acquisition (DAQ) systems, these can be prohibitively expensive and may not be suitable for real-time analysis. The cost of one of these data acquisition systems from manufactures such as Headwall, Resonon, or HySpex is not published but is commensurate with the cost of their hyperspectral imagers. The total cost of our data acquisition system is shown in the bill of materials table (Appendix A) and is estimated to be around an order of magnitude more affordable.

Our data acquisition system uses the open-source hyperspectral imager (OpenHSI), which is a compact and lightweight alternative that can be interfaced with development boards [56]. An open-source initiative lowers the barrier to entry for remote sensing practitioners to extend lab-based applications to the field. It also enables contributors to the project to become more involved in the fundamentals of hyperspectral imaging, including calibration, imagery alignment, and spectral data processing, because there are no proprietary barriers or a fixed system design and settings. The data acquisition system and OpenHSI camera combination is a self-contained solution that can be deployed on drones, aircraft, gantries, and tripods. Compared to pre-existing systems, this relies only on commercial-off-the-shelf (COTS) components, a PCB that can be ordered, and 3D printing.

Furthermore, a custom data acquisition solution offers opportunities to interact with sensor data during collection.

The work described in this paper is the first attempt at producing a true open-source solution for obtaining remotely sensed hyperspectral data. In this paper, we describe the design and methods needed to replicate the system. We include instructions for the build and operation and conclude with some initial results of our in-flight tests. We tested our DAQ and hyperspectral imager on a drone flight in Sydney, Australia, showing that hyperspectral, navigation, and orientation data can be collected and timestamped concurrently.

2. The Data Acquisition System

The OpenHSI camera and data acquisition system is a self-contained package that is designed to be deployed on a drone or aircraft [56] and is capable of collecting the camera position and orientation during flight for later georeferencing. Since the hyperspectral camera is a pushbroom sensor, the drone's forward motion provides the second spatial dimension. The specifications for the camera are shown in Table 1. The calibration and field validation of the OpenHSI camera design are described in [56]. Iterating on the original design, we used a FLIR detector for the data collected in this work. The open-source Python library used to operate the OpenHSI cameras works with a variety of detector manufacturers, including XIMEA, FLIR, and LUCID.

Table 1. Specifications for the OpenHSI camera.

Spectral range	430–830 nm
Spectral bands	213
Spatial resolution	5 cm at 120 m altitude
Cross-track pixels	800
Field of view	11°

The data acquisition system is arranged in a stack consisting of a compute board (Raspberry Pi 4B or NVIDIA Jetson), an uninterruptible power supply (UPS) and onboard batteries, and a Printed Circuit Board (PCB) containing the necessary ancillary sensors for direct georeferencing. Connected to the compute board by USB cables are a fast Solid State Device (SSD) for storage and the OpenHSI camera. (For the NVIDIA Jetson version, an M.2 slot SSD can be used instead). Through the General Purpose Input/Output (GPIO) pins, the ancillary sensor data are recorded in parallel with the hyperspectral data. A latching button is used to control the data collection, and LEDs indicate the system's status. Sensor data are processed and timestamped by a Teensy 4.0 microcontroller before being sent over UART to be saved by the Raspberry Pi. If desired, this datastream can be processed in real-time.

Raspberry Pis have been used in other remote sensing data acquisition tasks. For example, Belcore et al. used an RGB and Infrared Raspberry Pi camera on a drone to capture multispectral imagery [57]. For other lab-based hyperspectral tasks, Nasila et al. used a Raspberry Pi Zero W to interface a custom hyperspectral imager to a wireless network [46].

We found the a Raspberry Pi 4B was sufficient for slow tasks due to the limitations in processing power and USB data bus speeds—a tradeoff for the low cost. This was confirmed by [35], who found that a Raspberry Pi was insufficient for their needs. Hasler et al. successfully deployed a comparable hyperspectral imager controlled by a Jetson TX2 on a drone [49] and collected timestamped sensor data using a similar ancillary sensor board [55] for direct georeferencing. While the authors provided some tips and code listings, the full source and documentation were not available.

The main features of our DAQ are

- Entirely self-contained, including power;
- Navigation and orientation data for georeferencing collected in parallel;
- Uses commercial off-the-shelf components and 3D printing;
- Completely open-source with all design files and code available;

- Can be deployed remotely on drones and aircraft; and
- An easy-to-use interface (power on, press a button, and go).

Compared to closed-source commercial hyperspectral imagers, in terms of cost, the complete OpenHSI camera and data acquisition system is at least an order of magnitude more affordable. We chose SparkFun and Adafruit for most of our components, because they are reputable suppliers of sensors who also open-source their PCB designs and supporting software. Furthermore, we used an Arduino-compatible microcontroller and built upon the existing open-source code available. While the hardware design described in this paper is specific to the Raspberry Pi, a Jetson board can be used as a drop-in replacement at a higher cost. The GPIO requirements and software are the same for either option.

2.1. Electrical Components

In order to make it easy to interface with existing Raspberry Pi hats (accessory boards), the stacks can be connected using the 40 pin GPIO header. All ancillary sensors are controlled by a Teensy 4.0 microcontroller. The reason for including this microcontroller, rather than using the Raspberry Pi directly, is threefold: first, to obtain accurate timestamped sensor data without the delay caused by the operating system scheduler; second, to reduce the overhead for the Raspberry Pi in reading sensor data; and third, we needed two hardware serial ports, whereas the Raspberry Pi only has one. When we initially tried to read the sensor data directly with the Raspberry Pi, the camera frame rate dropped by more than half. The second hardware serial port allowed us to send selective diagnostic messages wirelessly using the XBee wireless radio module.

The board contains a GNSS module, a pressure/humidity/temperature sensor, a real time clock, connectors for the IMU, which is mounted on the camera, a wireless radio, and connectors for a latching button that controls data collection. There are also LED status indicators and a reset button to manually restart the microcontroller. The microcontroller runs a cooperative scheduler that schedules each sensor-read task at its desired frequency. Capacitors are used to stabilise the power and ground planes.

Each component is mounted on pin headers, and each PCB layer stacked, so that users can swap and remove modules as needed for a variety of deployment scenarios and applications. For example, if the XBee is physically removed for a drone flight, the rest of the software still runs but no longer initialises and schedules the Xbee task. This design also includes an independent uninterruptible power supply with 18,650 batteries, removing the reliance on buck converters and the host vehicles' (drones) power, which will reduce the flight time. Using the onboard USB-C port, the batteries can be charged, with an LED bar indicating the charge level.

A complete bill of materials for our DAQ system can be found in Appendix A. While these components can be sourced from the listed supplier, it is often more convenient to source them locally. Here in Australia, we purchased all of the components listed in Table A1 other than the UPS, SSD, and custom PCB from [58].

Resistors and capacitors are common components to buy in bulk, which reduces the cost for a project that only uses a few. With the PCBs, other fabrication houses will be suitable, but we used the suppliers JLCPCB and OSHPark. We chose black soldermask, but note that certain colours (green) will have faster turnaround times.

Since the data acquisition system and camera are designed to be deployed in a Ronin gimbal (or fixed mounted), the GNSS antenna needs to be brought out and fixed to the top of the drone to provide a direct line of sight. The displacement from the GNSS chip should be measured. The same microcontroller code can be used for the Sparkfun GPS RTK2 if RTK functionality is desired.

We chose the Teensy 4.0 microcontroller because of its many hardware serial ports and compatibility with the Adafruit and Sparkfun sensor libraries. We decided to mount every sensor in headers for ease of replacement and development. This proved to be helpful, as we accidentally blew some capacitors during testing.

The XBEE 3, XBEE USB, and XBEE Breakout listed in Table A1 are optional components. These components form a wireless serial pair, so the the DAQ status can be viewed in real-time during operation. Unfortunately, these wireless modules can only be used in the lab, since the signal is overpowered by the Matrice M600 drone when powered on. This can be rectified using other radios that operate at a different frequency.

The Raspberry Pi 4B, battery hat, and ancillary sensor board are stacked and mounted within the 3D printed housing. Figure 1 shows the assembled DAQ.

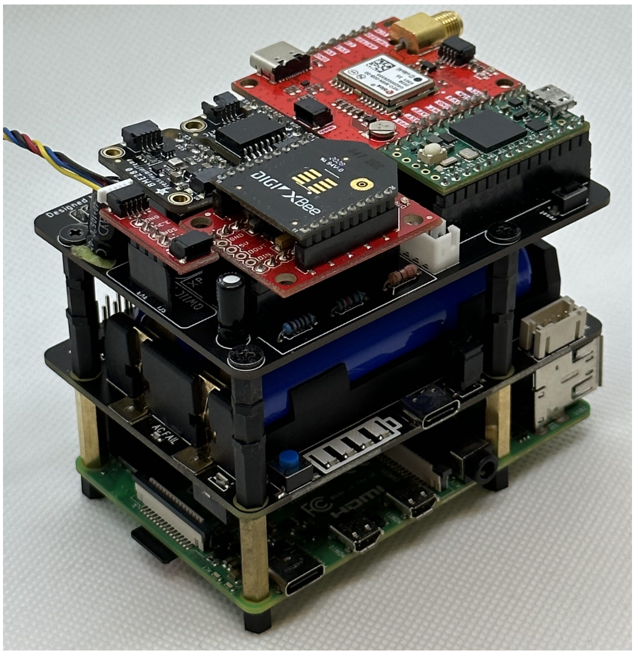
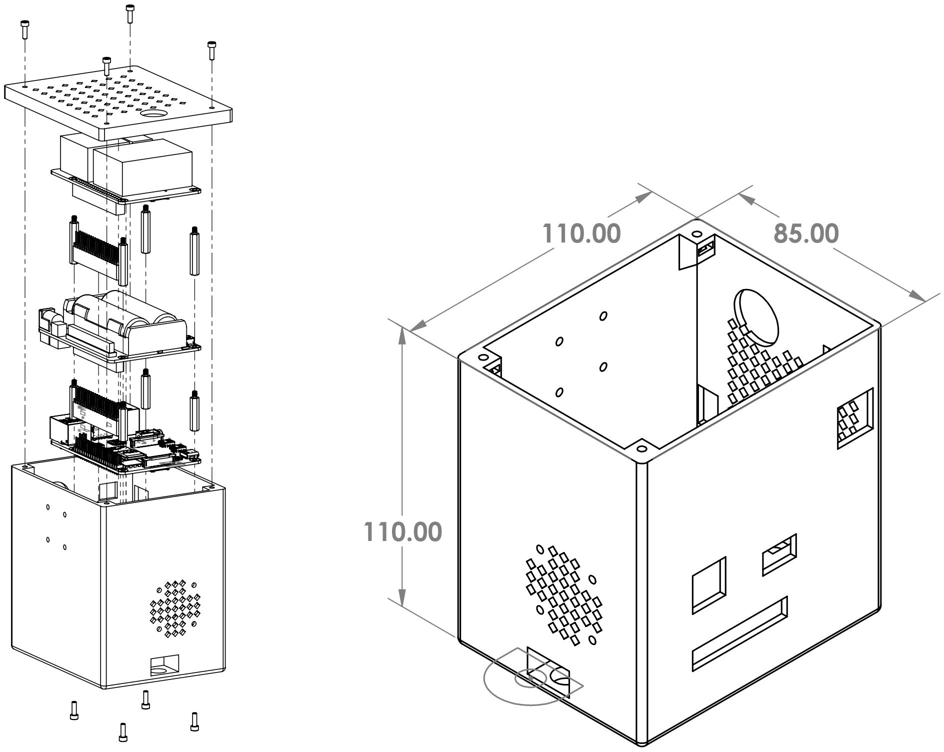


Figure 1. The assembled DAQ including the ancillary sensors. The IMU is connected via a cable so it can be mounted on the camera. The mass is 583 g including the SSD and camera, but not including the enclosure.

2.2. Design Files and Assembly

Each component for our DAQ is a commercial-off-the-shelf product and can be either ordered or 3D-printed. For the PCB, we provide the design files needed to order them from PCB fabrication houses. All design files and code can be found at [59]. The entire data acquisition system is mounted within a 3D-printed enclosure with holes for attaching mounting brackets and for airflow. To obtain the appropriate strength, our printer used a wall size of 2 mm and 100% infill.

Assembly is straightforward and involves stacking the Raspberry Pi 4B, battery hat, and ancillary sensor board with spacers to fix the structure in place. The battery hat includes the spacers needed, and we used some Adafruit nylon spacers for the ancillary sensor board. After the stack is complete, the entire system can be fixed to the 3D-printed enclosure with some standoffs. In Figure 2a, a visualisation of the assembly is shown. The overall dimensions of the housing are shown in Figure 2b. The enclosure also has holes to mount it directly to a dovetail bracket, allowing it to be deployed on a Ronin gimbal for our drone flights. Using some foam padding, we placed the SSD inside and connected it via USB. On the side of the 3D-printed enclosure are mounting points for attaching a bracket and the OpenHSI camera. The IMU is attached to a bracket and mounted on the side of the OpenHSI camera. Since the drone body interferes with the GNSS signal, the antenna needs to be carefully routed and fixed to the top of the drone.



(**a**) (**b**)

Figure 2. CAD designs of the assembly and enclosure. (**a**) Exploded view showing the assembly. (**b**) Isometric view of the enclosure showing cutouts for peripheral connections. The dimensions are 110 mm × 110 mm × 85 mm.

The entire setup weighed 735 g as mounted on a drone. There were a few things that could reduce the weight, such as using an SSD without the enclosure for the compute boards with an included M.2 slot. The battery hat could be removed if power could be sourced from the drone or aircraft. The ancillary sensors could also be soldered directly on, as opposed to using headers. This would reduce the weight down to 515 g. This payload is light enough for the DJI Phantom 4 carry capacity.

Once the hardware procedure is built, some software tweaks are needed. Some of the key steps are to enable GPIO, UART, and I2C and then add the current user to the `dialout` group. We also found it helpful to increase the USB memory limit and use PI0UART. The battery hat (X728) also has its own software [60] that needs to be installed. Once this was complete, we ran a script on power up using `systemd`. Similar steps are needed to install the software for the Jetson series of development boards. For more details on the software installation, visit our installation guide on our documentation website [61].

3. Operating Instructions

Before the DAQ can be deployed remotely, the batteries need to be charged. In our testing, the DAQ was capable of at least 2 h of data collection, which is significantly longer than the drone flight.

Although not necessary, we found that a gimbal greatly improved the data quality on our drone flights. The starting orientation of the camera should be recorded when the DAQ is powered on as this is needed for georectification. The camera can be assumed to be pointing directly down and rotated to a particular heading from geographic north.

A latching button (with an internal LED) is used to control the operation of the DAQ and camera, and a separate button powers the DAQ on and off. This interface was intentionally designed to be as simple as possible. LED status indicators provide information about the operation of the Raspberry Pi and ancillary sensor board stack.

Before takeoff, a GNSS fix and the IMU calibration can be checked by observing the heartbeat LED blinking faster than 1 Hz. A GNSS fix can take a few minutes from a cold start. This is summarised in Table 2. For more detailed sensor information, a laptop can parse the XBEE wireless messages in real time and show plots of sensor data on a dashboard. Unfortunately, the M600 drone drowned out the signal when powered up so this could only be used for preflight testing.

Table 2. A data collection status LED and heartbeat LED indicating the operational status.

LED	Activity	Meaning
Button LED	off	Data collection not in progress
Button LED	flashing	camera being initialised or busy
Button LED	solid	data being collected
Microcontroller LED	not blinking	code hung—hit the reset
Microcontroller LED	blinking at 1 HZ	code runs as normal but missing sufficient GNSS in view
Microcontroller LED	blinking at faster than 2 Hz	Data collection with enough GNSS satellites in view

After the DAQ is powered on, pressing the LED button begins the data collection, and depressing the button stops the data collection. When the LED indicator stops blinking, the navigation and hyperspectral data have finished saving to an onboard SSD. At this point, the SSD cable can then be unplugged from the Raspberry Pi or Jetson, and summary plots can be viewed to confirm the data were collected correctly.

The operating procedures are summarised in Figure 3.

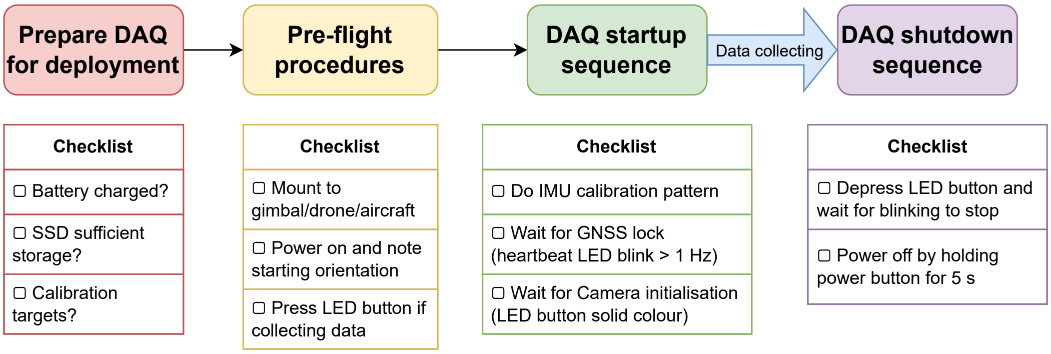

Prepare DAQ for deployment
Pre-flight procedures
DAQ startup sequence
Data collecting
DAQ shutdown sequence
Checklist
□ Battery charged?
□ SSD sufficient storage?
□ Calibration targets?
Checklist
□ Mount to gimbal/drone/aircraft
□ Power on and note starting orientation
□ Press LED button if collecting data
Checklist
□ Do IMU calibration pattern
□ Wait for GNSS lock (heartbeat LED blink > 1 Hz)
□ Wait for Camera initialisation (LED button solid colour)
Checklist
□ Depress LED button and wait for blinking to stop
□ Power off by holding power button for 5 s

Figure 3. Checklists for each stage of DAQ deployment.

For spectral validation and empirical line calibration, known spectral targets are beneficial. These can be calibration tarps or colour panels. A combination of a dark and a bright target with known spectral measurements is best. For verification of the georeferencing, a Trimble (or another comparable device) can be used to mark the tarp corners and other targets of interest in view.

There are a few considerations needed for obtaining high quality hyperspectral data. In order to obtain square pixels, the drone or aircraft needs to be flown at a specific speed. If this is not achieved, each pixel will be elongated into a rectangle. The parameters that control the flight speed are the camera frame rate and altitude.

Obtaining Square Pixels

The frame rate or frames-per-second (FPS) depends heavily on the capability of the data acquisition device. The USB throughput of the Raspberry Pi is a limiting factor for the DAQ, and because it is multiplexed, one cannot operate the camera and save the data to the SSD concurrently. However, this may not necessarily be an issue for some applications. We found the Jetson to be a superior compute platform since it does not have this issue, and hyperspectral data can be captured and saved with minimal latency. In this application, the fastest frame rate we could achieve on a Raspberry Pi was 25 FPS, so we used an integration time of 30 ms to maximise the signal-to-noise ratio without slowing the frame rate any further.

At an altitude l of 120 m and a cross track angular resolution δ of 0.24 mrad, the ground sample distance (GSD) is approximately

$$\text{GSD} = l\tan(\delta) \approx l\delta \approx 2.9 \text{ cm}. \tag{1}$$

Therefore, in order to image square pixels, the drone or aircraft needs to fly at a speed of

$$v = \text{GSD} \times \text{FPS} \approx 72 \text{ cm/s}. \tag{2}$$

See Figure 4 for a visual. Whether the drone or aircraft is able to fly at this speed depends on the platform and control parameters. For example, when using the Pix4D application to control the drone's flight path, the smallest allowable speed was 1 m/s, so we flew at 1.4 m/s (an integer multiple of v) and later spatially binned the data to 5.8 cm pixels. For higher spatial resolutions, the altitude l can be decreased accordingly.

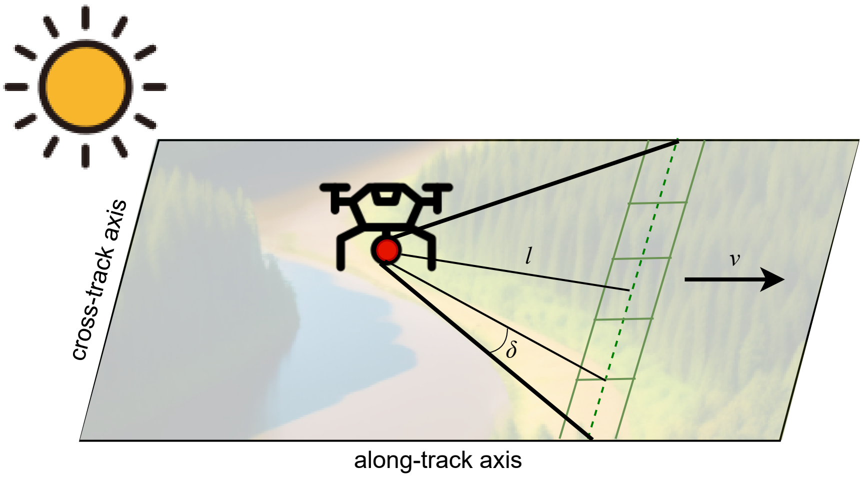


Figure 4. The drone payload, represented by the red disc, includes the data acquisition system and a pushbroom hyperspectral imager. As the drone flies, a swath is imaged. The forward motion provides the along-track axis. The spatial resolution is determined by the cross-track angular resolution δ and flight speed v.

4. Validation and Characterisation

We validated the DAQ on an Matrice M600 drone and collected hyperspectral data and navigation data simultaneously. Figure 5 shows the beach area where we setup tarps and cones to image with our completed DAQ mounted on a Ronin gimbal and deployed on a Matrice M600 drone.

A summary of the data is shown in Figure 6. From the quaternions, the camera orientation was calculated in a local north-east-down (NED) frame. After every data collection session, the timestamped sensor measurements were saved, along with summary plots.

Figure 7 demonstrates the Raspberry Pi 4B recording the UTC timestamps for each line recorded by the hyperspectral imager during the drone's forward motion. These timestamps were synced to the microcontroller's GNSS time using the pulse-per-second signal.

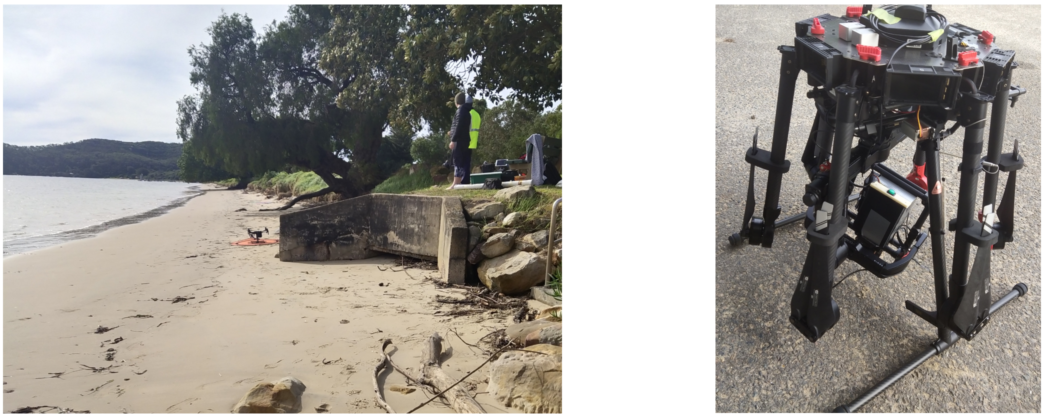


Figure 5. Trial site preparations. (**left**) The beach where we performed our in flight tests. (**right**) The completed DAQ and camera mounted on a gimbal ready to be deployed. The GNSS antenna was carefully routed to the top.

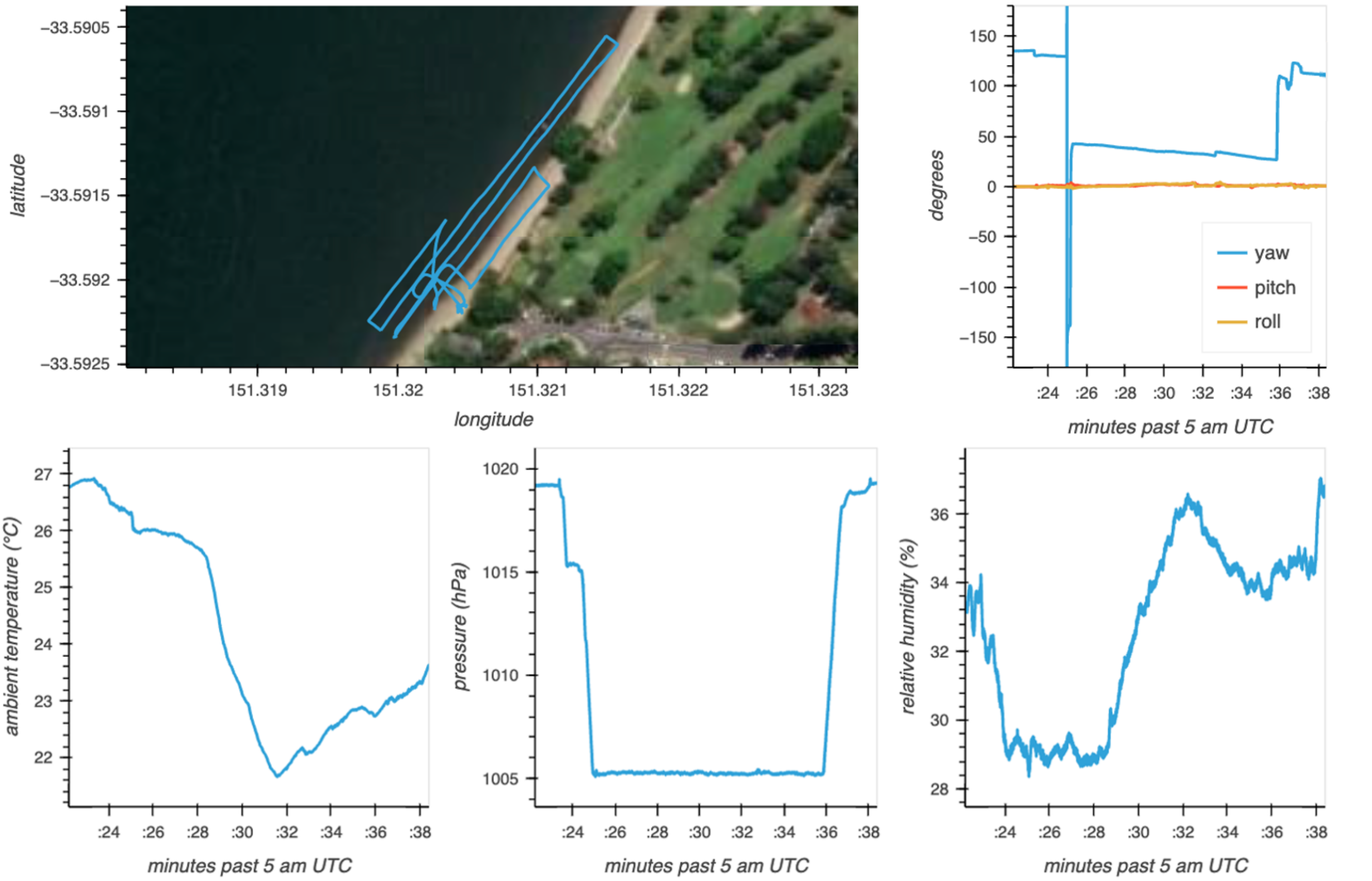


Figure 6. Summary plots of the navigation data showing the flight path, camera orientation in NED, and some atmospheric parameters. GNSS data were collected at 1 Hz and the rest at 100 Hz, which is faster than the camera frame rate.

The navigation and orientation data collected after time synchronisation were used to directly georeference the hyperspectral data collected. The resulting datacube can be visualised in Figure 8.

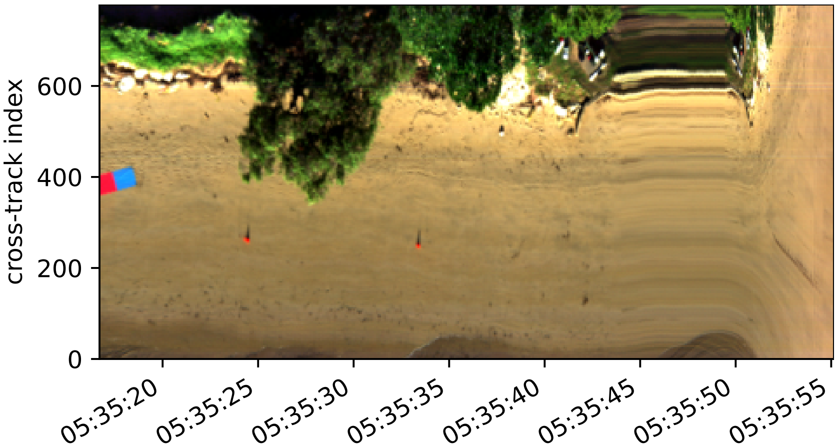


Figure 7. RGB composite of one hyperspectral datacube using selected bands (red = 640 nm, green 550 nm, blue 470 nm). The along-track axis is indexed by UTC time synced to the GNSS time. The stretch feature at 05:35:45–50 was due to the drone stopping before descending.

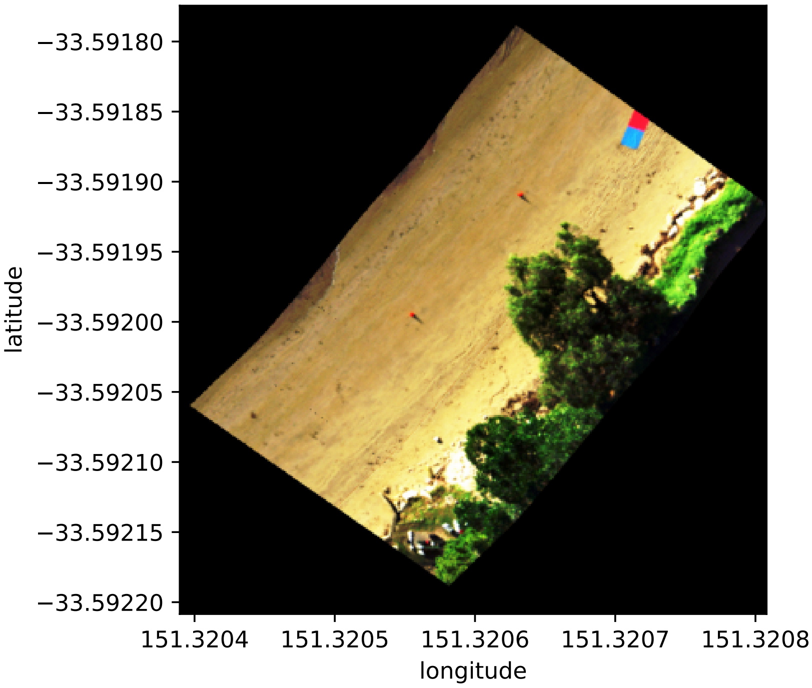


Figure 8. RGB composite (red = 640 nm, green 550 nm, blue 470 nm) of one hyperspectral datacube after georeferencing.

Usage Tips and Future Work

Without an RTK, the GNSS points were accurate to 1 m, and the maximum drift was observed while the data acquisition system was stationary. While this may not be an issue, depending on the spatial resolution required, the OpenHSI camera had a ground sampling distance of 5 cm, so this was significant. For the highest georeferencing accuracy, an RTK module is recommended. Future trials should collect ground control points so that the accuracy of the georeferenced positions of the cones and tarpaulins can be quantified.

At first startup, the data acquisition system requires time for a GNSS fix and IMU calibration. This is needed before takeoff and is indicated by an LED. Once the IMU has been calibrated, the starting orientation of the camera should be noted, as it is used as the initial condition for georectification. We found gimbal stabilisation helpful for increasing the quality of the georeferenced hyperspectral data, but it is not strictly necessary.

While we validated the data acquisition system on a successful flight, it is worth testing the performance under different wind conditions and without gimbal stabilisation. Doing so will inform users about the structural dynamics and suitability of the ancillary sensor data. This may mean setting constraints on deployment conditions. For example, if there are significant vibrations above 50 Hz or the hyperspectral imager needs to operate at s higher than 50 Hz frame rate, a different IMU with a higher than 100 Hz sample rate is needed.

Despite being affordable, the Raspberry Pi 4B has a multiplexed USB bus, so camera data pause while writing to a portable SSD using the same USB bus. Knowing that there is minimal latency during hyperspectral data collection would give more control over how to best image the area of interest and plan swath overlaps. We found the NVIDIA Jetson boards to be superior computing device, and our open-source software was designed to be compatible with these. Please see [61] for an installation guide. Other options include the Banana Pi, which has options for an M.2 SSD slot and more RAM. These options forgo the portable SSD in favour of an installed SSD module, thereby reducing the weight and overall size of the data acquisition system.

5. Conclusions

Previous uses of compact do-it-yourself hyperspectral imagers required the camera to be tethered, limiting the scope to lab-based or handheld applications. For remote sensing applications, a self-contained data acquisition system that is capable of recording hyperspectral and navigation data simultaneously is necessary. This work fills the gap and introduces a compact and customisable open-source data acquisition system for collecting time synchronised hyperspectral data alongside navigation and orientation data for direct georeferencing.

Our hardware design and software solution use only COTS components, a PCB that can be ordered, and a 3D-printed enclosure. A core design decision was to use easily replaceable components and allow ease of operation at the cost of some weight and height. The entire assembly weighed 735 g. We used a Raspberry Pi 4B to control the OpenHSI camera, and we found that the USB bus contention limited the maximum frame rate. A slightly more costly solution was to use a Jetson as a drop-in replacement.

By releasing our data acquisition system designs and code to the open-source community, we have lowered the barrier of entry for remote sensing practitioners to produce their own georeferenced hyperspectral imagery. This also enables contributors to be more involved in the fundamentals of hyperspectral imaging, including calibration, imagery alignment, spectral data processing, and georectification, because there are no proprietary barriers. With this work, we offer a more affordable option for the global community to build an entirely open-source data acquisition system and hyperspectral imager for remote sensing using commercial-off-the-shelf components. Overall, we demonstrated our open-source hardware design and software on a drone flight over Sydney, Australia, proving that hyperspectral remote sensing can be possible at low cost for everyone.

Author Contributions: Conceptualisation, Y.M., C.H.B., B.J.E., S.G. and K.C.W.; methodology, Y.M.; software, Y.M. and C.H.B.; validation, Y.M., C.H.B. and J.R.; formal analysis, Y.M.; investigation, Y.M., S.G. and J.R.; resources, Y.M., S.G., J.R., K.C.W. and C.H.B.; data curation, Y.M. and C.H.B.; writing—original draft preparation, Y.M.; writing—review and editing, Y.M., S.G., B.J.E. and I.H.C.; visualisation, Y.M.; supervision, C.H.B., J.R., B.J.E. and I.H.C.; project administration, B.J.E.; funding acquisition, B.J.E. and I.H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was conducted by the ARC Training Centre for Cubesats, UAVs, and their Applications (CUAVA—project number IC170100023) and funded by the Australian Government.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Available upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
COTS	commercial-off-the-shelf
CUAVA	Cubesats, UAVs and their Applications
DAQ	data acquisition
EO-1	Earth Observing-1
FPS	frames-per-second
GNSS	global navigation satellite system
GPIO	general purpose input/output
GSD	ground sample distance
I2C	inter-integrated circuit
IMU	inertial measurement unit
LED	light emitting diode
NED	north-east-down
OpenHSI	open-source hyperspectral imager
PCB	printed circuit board
PPS	pulse per second
RAM	random access memory
RGB	red green blue
SSD	solid state device
UART	universal asynchronous receiver transmitter
UPS	uninterruptable power supply
USB	universal serial bus
UTC	universal time coordinated

Appendix A. Bill of Materials

All the components needed to assemble a complete data acquisition system is listed in Table A1. The open-source hyperspectral imager can be built using the components listed in [56].

Table A1. Bill of materials. Total cost was $630 USD at the time of writing (sources all accessed on 24 July 2023) and did not include shipping or import duties.

Designator	Component	Qty	Unit Cost	Source of Materials
Raspberry Pi 4B	Dev board (4 GB or 8 GB)	1	$55+	raspberrypi.com/products/
SD Card	Storage 16 GB	1	$19.95	adafruit.com/product/2693
Fan	Minature 5V Cooling fan	1	$3.50	adafruit.com/product/3368
Heat sinks	Heat Sinks for Raspberry Pi	2	$1.50	adafruit.com/product/3083
Battery Hat	Geekworm UPS X728 V2.3	1	$45.99	geekworm.com
Battery	LiPo 18650 Battery Cells	2	$7.99	microcenter.com
Samsung T7	1TB Portable Solid State Drive	1	$89.99	samsung.com/us/
Teensy 4.0	PJRC Microcontroller	1	$23.80	adafruit.com/product/4323
USB cable	USB A to USB C	1	$2.95	adafruit.com/product/4472
GNSS	SparkFun NEO-M9N GPS (SMA)	1	$74.95	sparkfun.com/products/17285
Antenna	GNSS Antenna (SMA)	1	$13.95	sparkfun.com/products/14986
BNO055	Inertial Measurement Unit	1	$29.95	adafruit.com/product/4646
DS3231	Real time clock	1	$13.95	adafruit.com/product/5188
CR1220	Coin Cell Battery 12.5 mm	1	$0.95	adafruit.com/product/380
BME280	Pressure/Humidity/Temperature	1	$14.95	adafruit.com/product/2652

Table A1. *Cont.*

Designator	Component	Qty	Unit Cost	Source of Materials
XBEE 3 [1]	XBee 3 Pro Module (PCB Antenna)	2	$30.95	sparkfun.com/products/15127
XBEE USB [1]	SparkFun XBee Explorer Dongle	1	$27.95	sparkfun.com/products/11697
XBEE Breakout [1]	SparkFun XBee Explorer Regulated	1	$11.95	sparkfun.com/products/11373
40 pin header	Stacking headers	3	$2.50	adafruit.com/product/2223
Spacers	Nylon Screw and Stand-off Set	1	$16.95	adafruit.com/product/3299
Sensors PCB	Custom Ancillary Sensors board	1	$42.75	oshpark.com
Button	Latching Illuminated Push Button	1	$1.50	adafruit.com/product/1443
Quiic breakout	Qwiic Breadboard Breakout Adapter	1	$1.95	adafruit.com/product/4527
Connectors	JST-XH Connector Kit (optional)	1	$24.95	adafruit.com/product/4423
I^2C Connectors	Qwiic JST SH 4-Pin Cable 400 mm	1	$1.50	adafruit.com/product/5385
Resistors	10 kΩ and 330 Ω	4	$0.99	microcenter.com
Capacitors	10 µF	10	$1.95	adafruit.com/product/2193
Button	Microcontroller reset button	1	$1.49	pololu.com/product/1400
Headers	Assorted headers for breakout boards	5	$1-2	adafruit.com/category/154
Solder	Lead free solder wire 50 g	1	$12.50	adafruit.com/product/2473
		Total cost	≈ $630 USD	

[1] Optional component.

Appendix B. Ancillary Sensors

A summary of each sensor used to obtain the ancillary data is listed in Table A2. We chose to use SparkFun and Adafruit as they are both reputable suppliers of quality sensors and provide open-source software.

During development, we found that these sensors had random noise much lower than the stated manufacturer accuracy in Table A2. When we placed the IMU and GNSS module stationary on a desk, the absolute orientation stayed constant while the position drifted within 1.5 m over the course of a few minutes. Since the timescale of our data collection happened within a few tens of seconds, these long-term drift errors were not apparent in the georeferenced data shown in Figure 8.

Table A2. Ancillary sensor accuracy and max sample rate as stated from the manufacturer.

Ancillary Data	Sensor	Accuracy	Units	Max Sample Rate (Hz)
Position (GNSS)	NEO-M9N	1.5	m	25
Absolute Orientation (IMU)	BNO055	1	deg	100
Time (RTC)	DS3231	3.5	µs	100+
Pressure	BME280	1	hPa	120 [1]
Humidity	BME280	3	%RH	120 [1]
Temperature	BME280	1	°C	120 [1]

[1] For all three measurements taken together. Sample rate depends on amount of oversampling and differs for measuring pressure, humidity, and temperature.

Since the GNSS library operates with a family of SparkFun GNSS modules, the GNSS modules can be hotswapped depending on the desired features such as an RTK, or choice of a U.FL or SMA connector. The PCB design accommodates two sizes corresponding to the SparkFun NEO-M9N with the SMA and with the Chip antenna. If the RTK version is used, the horizontal position accuracy is 0.01 m but costs almost four times more. These GNSS modules can take up to 30 s for a cold fix so during this time, one can perform the calibration pattern for the IMU.

The IMU used is the Adafruit BNO055 which has some features that reduce the complexity required to obtain an orientation solution. This IMU was chosen for its ability to output absolute orientation by sensor fusion of the internal gyroscope, accelerometer, and magnetometer. While the output can be Euler angles, we chose to record quaternions as these can be spherically interpolated according to the camera timestamps.

A temperature compensated RTC is used to keep track of timestamps for each sensor update. This is necessary for later synchronisation with the hyperspectral data. Depending on the timescale of the data collection, the DS1307 RTC is a possible replacement. However, it is not temperature compensated (unlike the DS3231) and drifts by a few seconds per day.

The BME280 measures pressure which can fine tune the estimated GNSS altitude. In addition, it is able to measure relative humidity so it can be used to estimate a portion of the water column as a drone rises, which may be useful for atmospheric modelling and radiative transfer.

References

1. Kruse, F.A.; Boardman, J.W.; Huntington, J.F. Comparison of airborne hyperspectral data and EO-1 Hyperion for mineral mapping. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1388–1400. [CrossRef]
2. Van der Meer, F.D.; Van der Werff, H.M.; Van Ruitenbeek, F.J.; Hecker, C.A.; Bakker, W.H.; Noomen, M.F.; Van Der Meijde, M.; Carranza, E.J.M.; De Smeth, J.B.; Woldai, T. Multi-and hyperspectral geologic remote sensing: A review. *Int. J. Appl. Earth Obs. Geoinf.* **2012**, *14*, 112–128. [CrossRef]
3. Vaughan, R.G.; Calvin, W.M.; Taranik, J.V. SEBASS hyperspectral thermal infrared data: Surface emissivity measurement and mineral mapping. *Remote Sens. Environ.* **2003**, *85*, 48–63. [CrossRef]
4. Rani, N.; Mandla, V.R.; Singh, T. Evaluation of atmospheric corrections on hyperspectral data with special reference to mineral mapping. *Geosci. Front.* **2017**, *8*, 797–808. [CrossRef]
5. Yokoya, N.; Chan, J.C.W.; Segl, K. Potential of resolution-enhanced hyperspectral data for mineral mapping using simulated EnMAP and Sentinel-2 images. *Remote Sens.* **2016**, *8*, 172. [CrossRef]
6. Chabrillat, S.; Goetz, A.F.; Krosley, L.; Olsen, H.W. Use of hyperspectral images in the identification and mapping of expansive clay soils and the role of spatial resolution. *Remote Sens. Environ.* **2002**, *82*, 431–445. [CrossRef]
7. Transon, J.; d'Andrimont, R.; Maugnard, A.; Defourny, P. Survey of hyperspectral earth observation applications from space in the sentinel-2 context. *Remote. Sens.* **2018**, *10*, 157. [CrossRef]
8. Mashimbye, Z.; Cho, M.; Nell, J.; De Clercq, W.; Van Niekerk, A.; Turner, D. Model-based integrated methods for quantitative estimation of soil salinity from hyperspectral remote sensing data: A case study of selected South African soils. *Pedosphere* **2012**, *22*, 640–649. [CrossRef]
9. Kaufmann, H.; Segl, K.; Chabrillat, S.; Hofer, S.; Stuffler, T.; Mueller, A.; Richter, R.; Schreier, G.; Haydn, R.; Bach, H. EnMAP a hyperspectral sensor for environmental mapping and analysis. In Proceedings of the 2006 IEEE International Symposium on Geoscience and Remote Sensing, Denver, CO, USA, 31 July–4 August 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 1617–1619.
10. He, K.S.; Rocchini, D.; Neteler, M.; Nagendra, H. Benefits of hyperspectral remote sensing for tracking plant invasions. *Divers. Distrib.* **2011**, *17*, 381–392. [CrossRef]
11. Schmidtlein, S.; Sassin, J. Mapping of continuous floristic gradients in grasslands using hyperspectral imagery. *Remote Sens. Environ.* **2004**, *92*, 126–138. [CrossRef]
12. Adam, E.; Mutanga, O.; Rugege, D. Multispectral and hyperspectral remote sensing for identification and mapping of wetland vegetation: A review. *Wetl. Ecol. Manag.* **2010**, *18*, 281–296. [CrossRef]
13. Underwood, E.; Ustin, S.; DiPietro, D. Mapping nonnative plants using hyperspectral imagery. *Remote Sens. Environ.* **2003**, *86*, 150–161. [CrossRef]
14. Andrew, M.E.; Ustin, S.L. The role of environmental context in mapping invasive plants with hyperspectral image data. *Remote Sens. Environ.* **2008**, *112*, 4301–4317. [CrossRef]
15. Lee, Z.; Carder, K.L.; Mobley, C.D.; Steward, R.G.; Patch, J.S. Hyperspectral remote sensing for shallow waters. I. A semianalytical model. *Appl. Opt.* **1998**, *37*, 6329–6338. [CrossRef]
16. Sandidge, J.C.; Holyer, R.J. Coastal bathymetry from hyperspectral observations of water radiance. *Remote Sens. Environ.* **1998**, *65*, 341–352. [CrossRef]
17. Keith, D.J.; Schaeffer, B.A.; Lunetta, R.S.; Gould, R.W., Jr.; Rocha, K.; Cobb, D.J. Remote sensing of selected water-quality indicators with the hyperspectral imager for the coastal ocean (HICO) sensor. *Int. J. Remote Sens.* **2014**, *35*, 2927–2962. [CrossRef]
18. Brando, V.E.; Dekker, A.G. Satellite hyperspectral remote sensing for estimating estuarine and coastal water quality. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1378–1387. [CrossRef]
19. Lee, Z.; Carder, K.L.; Mobley, C.D.; Steward, R.G.; Patch, J.S. Hyperspectral remote sensing for shallow waters: 2. Deriving bottom depths and water properties by optimization. *Appl. Opt.* **1999**, *38*, 3831–3843. [CrossRef] [PubMed]
20. Dozier, J.; Painter, T.H. Multispectral and hyperspectral remote sensing of alpine snow properties. *Annu. Rev. Earth Planet. Sci.* **2004**, *32*, 465–494. [CrossRef]
21. Hulley, G.C.; Duren, R.M.; Hopkins, F.M.; Hook, S.J.; Vance, N.; Guillevic, P.; Johnson, W.R.; Eng, B.T.; Mihaly, J.M.; Jovanovic, V.M.; et al. High spatial resolution imaging of methane and other trace gases with the airborne Hyperspectral Thermal Emission Spectrometer (HyTES). *Atmos. Meas. Tech.* **2016**, *9*, 2393–2408. [CrossRef]
22. Crevoisier, C.; Nobileau, D.; Fiore, A.M.; Armante, R.; Chédin, A.; Scott, N. Tropospheric methane in the tropics—First year from IASI hyperspectral infrared observations. *Atmos. Chem. Phys.* **2009**, *9*, 6337–6350. [CrossRef]

23. Calin, M.A.; Calin, A.C.; Nicolae, D.N. Application of airborne and spaceborne hyperspectral imaging techniques for atmospheric research: Past, present, and future. *Appl. Spectrosc. Rev.* **2021**, *56*, 289–323. [CrossRef]
24. Montmessin, F.; Gondet, B.; Bibring, J.P.; Langevin, Y.; Drossart, P.; Forget, F.; Fouchet, T. Hyperspectral imaging of convective CO2 ice clouds in the equatorial mesosphere of Mars. *J. Geophys. Res. Planets* **2007**, *112*. [CrossRef]
25. Williams, D.J.; Feldman, B.L.; Williams, T.J.; Pilant, D.; Lucey, P.G.; Worthy, L.D. Detection and identification of toxic air pollutants using airborne LWIR hyperspectral imaging. In Proceedings of the Multispectral and Hyperspectral Remote Sensing Instruments and Applications II, Honolulu, HI, USA, 9–11 November 2005; Volume 5655, pp. 134–141.
26. Pearlman, J.S.; Barry, P.S.; Segal, C.C.; Shepanski, J.; Beiso, D.; Carman, S.L. Hyperion, a space-based imaging spectrometer. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1160–1173. [CrossRef]
27. Folkman, M.A.; Pearlman, J.; Liao, L.B.; Jarecke, P.J. EO-1/Hyperion hyperspectral imager design, development, characterization, and calibration. *Hyperspectral Remote Sens. Land Atmos.* **2001**, *4151*, 40–51.
28. Datt, B.; McVicar, T.R.; Van Niel, T.G.; Jupp, D.L.; Pearlman, J.S. Preprocessing EO-1 Hyperion hyperspectral data to support the application of agricultural indexes. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 1246–1259. [CrossRef]
29. Green, R.O.; Eastwood, M.L.; Sarture, C.M.; Chrien, T.G.; Aronsson, M.; Chippendale, B.J.; Faust, J.A.; Pavri, B.E.; Chovit, C.J.; Solis, M.; et al. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sens. Environ.* **1998**, *65*, 227–248. [CrossRef]
30. Vane, G.; Green, R.O.; Chrien, T.G.; Enmark, H.T.; Hansen, E.G.; Porter, W.M. The airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sens. Environ.* **1993**, *44*, 127–143. [CrossRef]
31. Chapman, J.W.; Thompson, D.R.; Helmlinger, M.C.; Bue, B.D.; Green, R.O.; Eastwood, M.L.; Geier, S.; Olson-Duvall, W.; Lundeen, S.R. Spectral and radiometric calibration of the next generation airborne visible infrared spectrometer (AVIRIS-NG). *Remote Sens.* **2019**, *11*, 2129. [CrossRef]
32. Kruse, F.; Boardman, J.; Lefkoff, A.; Young, J.; Kierein-Young, K.; Cocks, T.; Jensen, R.; Cocks, P. HyMap: An Australian hyperspectral sensor solving global problems-results from USA HyMap data acquisitions. In Proceedings of the 10th Australasian Remote Sensing and Photogrammetry Conference, Adelaide, Australia, 25 August 2000; pp. 18–23.
33. Schlerf, M.; Atzberger, C.; Hill, J. Remote sensing of forest biophysical variables using HyMap imaging spectrometer data. *Remote Sens. Environ.* **2005**, *95*, 177–194. [CrossRef]
34. Huang, Z.; Turner, B.J.; Dury, S.J.; Wallis, I.R.; Foley, W.J. Estimating foliage nitrogen concentration from HYMAP data using continuum removal analysis. *Remote Sens. Environ.* **2004**, *93*, 18–29. [CrossRef]
35. Wang, L.; Jin, J.; Song, Z.; Wang, J.; Zhang, L.; Rehman, T.U.; Ma, D.; Carpenter, N.R.; Tuinstra, M.R. LeafSpec: An accurate and portable hyperspectral corn leaf imager. *Comput. Electron. Agric.* **2020**, *169*, 105209. [CrossRef]
36. Rascher, U.; Nichol, C.J.; Small, C.; Hendricks, L. Monitoring spatio-temporal dynamics of photosynthesis with a portable hyperspectral imaging system. *Potogrammetric Eng. Remote Sens.* **2007**, *73*, 45–56. [CrossRef]
37. Yao, X.; Cai, F.; Zhu, P.; Fang, H.; Li, J.; He, S. Non-invasive and rapid pH monitoring for meat quality assessment using a low-cost portable hyperspectral scanner. *Meat Sci.* **2019**, *152*, 73–80. [CrossRef]
38. Ogawa, Y.; Tanaka, Y.; Kadokura, A.; Hosokawa, K.; Ebihara, Y.; Motoba, T.; Gustavsson, B.; Brändström, U.; Sato, Y.; Oyama, S.; et al. Development of low-cost multi-wavelength imager system for studies of aurora and airglow. *Polar Sci.* **2020**, *23*, 100501. [CrossRef]
39. Chen, J.; Cai, F.; He, R.; He, S. Experimental demonstration of remote and compact imaging spectrometer based on mobile devices. *Sensors* **2018**, *18*, 1989. [CrossRef]
40. Oh, S.W.; Brown, M.S.; Pollefeys, M.; Kim, S.J. Do it yourself hyperspectral imaging with everyday digital cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2461–2469.
41. Henriksen, M.B.; Prentice, E.F.; Van Hazendonk, C.M.; Sigernes, F.; Johansen, T.A. Do-it-yourself vis/nir pushbroom hyperspectral imager with c-mount optics. *Opt. Contin.* **2022**, *1*, 427–441. [CrossRef]
42. Sigernes, F.; Syrjäsuo, M.; Storvold, R.; Fortuna, J.; Grøtte, M.E.; Johansen, T.A. Do it yourself hyperspectral imager for handheld to airborne operations. *Opt. Express* **2018**, *26*, 6021–6035. [CrossRef] [PubMed]
43. Ji, Y.; Tan, F.; Zhao, S.; Feng, A.; Zeng, C.; Liu, H.; Wang, C. Spatial-spectral resolution tunable snapshot imaging spectrometer: analytical design and implementation. *Appl. Opt.* **2023**, *62*, 4456–4464. [CrossRef]
44. Davies, M.; Stuart, M.B.; Hobbs, M.J.; McGonigle, A.J.; Willmott, J.R. Image correction and In situ spectral calibration for low-cost, smartphone hyperspectral imaging. *Remote Sens.* **2022**, *14*, 1152. [CrossRef]
45. Arad, O.; Chepelnov, L.; Afgin, Y.; Reshef, L.; Brikman, R.; Elatrash, S.; Stern, A.; Tsror, L.; Bonfil, D.J.; Klapp, I. Low-cost dispersive hyperspectral sampling scanner for agricultural imaging spectrometry. *IEEE Sens. J.* **2023**, *23*, 18292–18303. [CrossRef]
46. Näsilä, A.; Trops, R.; Stuns, I.; Havia, T.; Saari, H.; Guo, B.; Ojanen, H.J.; Akujärvi, A.; Rissanen, A. Hand-held MEMS hyperspectral imager for VNIR mobile applications. In Proceedings of the MOEMS and Miniaturized Systems XVII, San Francisco, CA, USA, 30–31 January 2018; Volume 10545, pp. 177–185.
47. Morales, A.; Horstrand, P.; Guerra, R.; Leon, R.; Ortega, S.; Díaz, M.; Melián, J.M.; López, S.; López, J.F.; Callico, G.M.; et al. Laboratory hyperspectral image acquisition system setup and validation. *Sensors* **2022**, *22*, 2159. [CrossRef]
48. Cocking, J.; Narayanaswamy, B.E.; Waluda, C.M.; Williamson, B.J. Aerial detection of beached marine plastic using a novel, hyperspectral short-wave infrared (SWIR) camera. *Ices J. Mar. Sci.* **2022**, *79*, 648–660. [CrossRef]

49. Hasler, O.; Løvås, H.; Bryne, T.H.; Johansen, T.A. Direct georeferencing for Hyperspectral Imaging of ocean surface. In Proceedings of the 2023 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2023; pp. 1–19.
50. Stuart, M.B.; McGonigle, A.J.; Willmott, J.R. Hyperspectral imaging in environmental monitoring: A review of recent developments and technological advances in compact field deployable systems. *Sensors* **2019**, *19*, 3071. [CrossRef]
51. Selva, D.; Krejci, D. A survey and assessment of the capabilities of Cubesats for Earth observation. *Acta Astronaut.* **2012**, *74*, 50–68. [CrossRef]
52. Wright, R.; Lucey, P.; Crites, S.; Horton, K.; Wood, M.; Garbeil, H. BBM/EM design of the thermal hyperspectral imager: An instrument for remote sensing of earth's surface, atmosphere and ocean, from a microsatellite platform. *Acta Astronaut.* **2013**, *87*, 182–192. [CrossRef]
53. Poghosyan, A.; Golkar, A. CubeSat evolution: Analyzing CubeSat capabilities for conducting science missions. *Prog. Aerosp. Sci.* **2017**, *88*, 59–83. [CrossRef]
54. Grøtte, M.E.; Birkeland, R.; Honoré-Livermore, E.; Bakken, S.; Garrett, J.L.; Prentice, E.F.; Sigernes, F.; Orlandić, M.; Gravdahl, J.T.; Johansen, T.A. Ocean color hyperspectral remote sensing with high resolution and low latency—The HYPSO-1 CubeSat mission. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 1–19. [CrossRef]
55. Albrektsen, S.M.; Johansen, T.A. User-configurable timing and navigation for UAVs. *Sensors* **2018**, *18*, 2468. [CrossRef]
56. Mao, Y.; Betters, C.H.; Evans, B.; Artlett, C.P.; Leon-Saval, S.G.; Garske, S.; Cairns, I.H.; Cocks, T.; Winter, R.; Dell, T. OpenHSI: A complete open-source hyperspectral imaging solution for everyone. *Remote Sens.* **2022**, *14*, 2244. [CrossRef]
57. Belcore, E.; Piras, M.; Pezzoli, A.; Massazza, G.; Rosso, M. Raspberry PI 3 multispectral low-cost sensor for UAV based remote sensing. Case study in south-west Niger. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 207–214. [CrossRef]
58. Core Electronics. Available online: https://core-electronics.com.au/ (accessed on 20 July 2023).
59. Hardware Files. Available online: https://github.com/openhsi/hardware_files (accessed on 20 July 2023).
60. X728. Available online: https://wiki.geekworm.com/X728 (accessed on 20 July 2023).
61. Installing OpenHSI on Linux. Available online: https://openhsi.github.io/openhsi/tutorials/installing_linux.html (accessed on 20 July 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.