INTERACTIVE HYPERSPECTRAL DATA INSPECTION DURING FIELD OPERATIONS

Dennis D. Langer[1], *Tor A. Johansen*[2], *Asgeir J. Sørensen*[1]

Centre for Autonomous Marine Operations and Systems
[1]Department of Marine Technology
[2]Department of Engineering Cybernetics
Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ABSTRACT

Hyperspectral data cubes generated from hyperspectral cameras can be hard to inspect in during a field operation as the data is collected. This is due to the high spectral dimension of the data, resulting in large data size. This paper proposes a method and interactive Software (SW) tool for fast monitoring of hyperspectral data cubes, enabling rapid decision making and mission re-planning in the field. Experimental data and lessons learned from field operations using Unmanned Aerial Vehicles (UAVs) equipped with hyperspectral cameras demonstrate the usefulness of the proposed method and SW tool.

***Index Terms*—** Hyperspectral, Mission re-planning, Data inspection, Interactive

1. INTRODUCTION

Hyperspectral data recorded from hyperspectral cameras are often considered as three dimensional in nature (two spatial and one spectral dimension), hence the term data *cube*. Hyperspectral data cubes are usually from hundreds of megabytes to tens of gigabytes large, containing millions of spectra, and can thus be hard to inspect in detail during the field operation due to operational and time constraints. For example, during a field test done by NTNU using a Specim AFX10 camera [1] on a multicopter drone, a 1.4 GB large data cube was generated with the relatively low frame rate of 20 Hz along a single flight line over the span of just two and a half minutes. The cube dimensions are 3201 lines by 1024 samples and 224 bands with data type 16 bit unsigned integer. The camera supports frame rates of over 100 Hz, potentially generating even larger data cubes.

Typical hyperspectral remote sensing cameras are pushbroom sensors, meaning a spatial line of pixels is recorded over a single exposure of the image sensor. A single exposure or frame of the 2D image sensor generates spatial-spectral data, as opposed to spatial-spatial data for conventional consumer cameras. A data cube is assembled by scanning multiple frames or lines across the scene to be measured. The workflow for planning a mission with such an imaging system can vary. One workflow can be to pick a desired ground resolution and flight altitude, and derive flight speed and frame rate from it. In addition, the sensor platform's continuous operation time capability (defined e.g. by battery capacity) and the hyperspectral instrument's coverage (swath width) need to be considered in some cases. Exposure time can be chosen depending on ambient light conditions influenced by weather and is constrained by the chosen frame rate. See [2] for an overview of how airborne remote sensing mission planning is done. Mission planning of airborne remote sensing surveys is subject to uncertainty, since weather and other hard to predict factors and limitations can induce operational constraints and force change of mission plans.

The spectral information is beyond the color perception capabilities of the human eye, and there exists many articles that discuss various methods for hyperspectral data visualization. Most of these papers present algorithms that post-process the data to maximize some measure of information contained in an Red-Green-Blue (RGB) composite representation of the cube [3, 4, 5, 6]. This is reasonable because the three channel RGB is the output mode of conventional computer displays. Some other papers [7, 8] discuss algorithms for optimal RGB band selection of the unprocessed data instead.

Many methods are based on Principal Component Analysis (PCA), and visualize the most significant components as RGB composites, such as [4] which presents a tool similar in interactivity as the one proposed here, where the user can in real time pick a blend of two bands for each RGB channel of a dimensionality reduced hyperspectral data cube. [3] summarizes well various tools and methods for hyperspectral data visualization, and describes an interactive MATLAB tool that enhances visual quality of an RGB composite. Recently also neural network based visualization is being considered [5].

This work was supported by the Research Council of Norway (RCN) through the MASSIVE project, grant number 270959, by the SeaBee infrastructure grant 296478, by the center of excellence AMOS, grant number 223254, and ELO-Hyp (Norway Grants contract 24/2020).

When visualizing hyperspectral data as RGB composites, the relatively small bitdepth of 8 bit per channel that a conventional computer display supports compared to 10, 12 or even 16 bit values of hyperspectral data can lead to substantial loss of details. One paper solves that issue by developing a method inspired by how High Dynamic Range (HDR) content is processed for display on conventional Standard Dynamic Range (SDR) displays [9].

Little published work exists discussing how real time monitoring or visualization of hyperspectral data can be done. One research team is discussing visualization of data during an operation using GoogleEarth [10], however, the paper only demonstrates the implementation of a small part of the proposed system in simulation and there is no follow-up study.

While lots of work has been done discussing how to visualize hyperspectral data cubes for analysis, they do not discuss the use of visualization or data inspection tools in an operational context, where time constraints may exists and ease-of-use plays an essential role, for example between survey flights of a multi-rotor drone, while replacing batteries and preparing the next flight. In such contexts, it can be advantageous to have a SW package or tool ready for use, that is sufficient for in-depth data inspection, especially when operating Do-It-Yourself (DIY) instruments. None of the cited methods are suitable when the raw data itself may be degraded in quality, which is a possibility with low-cost Commercial Off-The-Shelf (COTS) DIY hyperspectral cameras like the ones described in [11, 12, 13] that were built in-house. DIY instruments may contain errors introduced during assembly, or have other mechanical or optical issues that degrade the data quality. Some class of defects may appear only during an operation in the field. Complex data processing can hide defects that could be discovered and handled already during the operation. A visualization method that shows raw data as is, without processing to enhance details, makes it easier to infer the correct image quality camera performance.

This paper's contribution is to present a method for real time hyperspectral data monitoring and fast hyperpsectral data cube inspection using a SW tool that focuses on simplicity, essential features and preserving the raw state of the data. This enables rapid and real-time mission re-planning for greater mission success. The performance of the proposed method is demonstrated through experimental data.

The next section presents the proposed methods and SW tool, the underlying reasoning for their design, interactivity and operations where they were used. Concluding remarks follow in the end.

2. METHOD

The proposed software builds on top of the LSTS toolchain [14], which is a modular SW framework for networked vehicle systems. The relevant parts of the LSTS toolchain are DUNE, IMC and Neptus. The proposed method consists of robotic platforms equipped with a hyperspectral camera and running DUNE, which is a run-time environment for embedded systems used to develop custom SW for communication, sensor control and other applications. The robotic platforms are communicating using the Inter Module Communication (IMC) protocol. The system is monitored and controlled using Neptus, which is a graphical user interface for command and control of robotic systems.

2.1. Real time monitoring

The real time monitoring of hyperspectral data is done by using two approaches differing in required communication bandwidth. The first approach is low bandwidth and is done by computing intensity statistics as indicators for overall light signal intensity in every spatial line. This is seen in the console output of DUNE[1] and enables real time determination of a suitable gain or exposure time, and thereby enables real time re-planning of frame rate and other related mission parameters.

The second approach requires more bandwidth. The DUNE instance running on the hyperspectral sensor platform transmits IMC protocol messages containing either one or three user selected bands from each spectrum of the recorded spatial line, which is visualized in the ground station as a monochrome or RGB waterfall plot. This enables the operator to monitor and verify spatial coverage.

2.2. Inspection

In addition to the real time approaches, a SW tool (see [15] for source code) for fast and interactive inspection is proposed. It can be run on a small laptop in the field. The hyperspectral data can be inspected as soon as the data files are available in the laptop's file system. It presents the user with three views, see Fig. 1. The bottom view is a spatial monochrome or spatial RGB composite, the top left view shows a selected single frame or a spatial-spectral subset (this subset may also be called a spectrogram) of the cube. Lastly, the top right view shows the spectrum of a selected pixel from the top left view.

The visualization method can be viewed as removing one dimension from the data cube at a time by selection of successive data slices, going from cube, to spectrogram or frame and to spectrum. In this way, no part of the data is removed as it is with methods that use projections onto feature spaces like PCA or band selection methods. While not all information is visible simultaneously, all parts are potentially viewable, and

[1] pending more implementation work, this could also be made available via IMC protocol

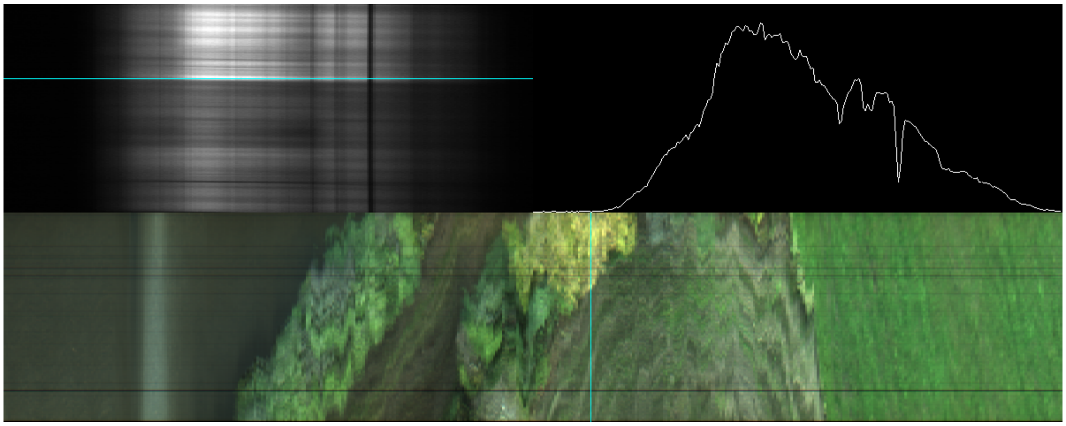


Fig. 1. The viewport of the proposed tool that is presented to the user. The data cube is from a drone flight using a Sigernes HSIv4 [11] in a field trip done during a hyperspectral remote sensing course at the Norwegian University of Science and Technology. The cyan lines in the bottom RGB composite indicate the selected spectrogram shown to the top left, and the cyan line in the top left indicate the selected spectrum shown to the top right. (The dark lines in the spatial composite are impurities in the entrance slit) See this YouTube video (https://youtu.be/JrK8qA3wW4k) for a short demonstration of the tool.

the user can quickly scan the cube and home into features of interest.

The issue with limited bit-depth of computer displays is avoided by displaying a line plot of a selected spectrum, mapping data cube values not into 255 levels of display intensity but instead to pixels available to the user's screen.

The proposed tool may also be used in the lab or for analysis. However, depending on the scientist's preference, completely custom tools like python scripts, packages like spectralpy, or SW suites like Matlab or ENVI may be preferred.

2.3. Interactivity

Neptus enables real time configuration of selected parameters of DUNE instances. Camera parameters like gain, frame rate and exposure time were made configurable in real time, and the user can choose to update them in response to the waterfall plot or intensity statistics. In addition, the user can choose to enable or disable both monitoring methods, and set a decimation factor N, to output information of only every Nth line for bandwidth control. The bands to select for the second approach is also user configurable.

The SW tool was designed to be fast and provide instantaneous feedback. The user selects a slice to show in the spectrogram view to the top left by clicking on the corresponding column in the monochrome or RGB composite view. A spectrum is selected by clicking on the corresponding row in the spectrogram view. The user can sweep through the lines or spectra of the data cube using arrow keys. The bands that are used for the RGB composite view can be changed, and RGB or monochromatic display of the cube can be set. The spatial composite can also be exported to an image file in .png format.

2.4. Implementation Details

The intensity indicators that are used in the low bandwidth approach are

1. The maximum intensity value in a line,
2. the 99.99 percentile intensity, meaning the intensity threshold below which 99.99% of all pixels in the line fall and
3. The percentage of pixels that are above 95% of the saturation intensity.

Wireless communication can be established using long range radio communication from the payload running DUNE to the ground station running Neptus. Data should be transferred using UDP as opposed to TCP, due to congestion avoidance procedures and re-transmissions as part of TCP can overload the communication link and introduce delays for links with high error rates [16].

The SW tool was developed in C++ using OpenGL [17] for rendering, SFML [18] for window management and user input, and LodePNG [19] for PNG file export. Currently supports data cubes in BIP format. BIL and BSQ format is pending implementation.

A limitation of the SW tool is that a cube has to fit into the computer's main memory. If not, a contiguous subset of the cube must be specified or performance is degraded.

2.5. Case where it could have been useful

During a mission using an AFX10 [1], it turned out that the last flight in a set of three, covering the same area did not cover one of the two reflectance targets that were placed for reflectance calibration, see Fig. 2. Had the tool been available, this error could have been discovered and corrected by noticing that the target was farther off center than expected and adjusting the flight path. Any tool that generates an RGB composite from the cube with reasonably ease and short processing time would have sufficed, however a visualization that is integrated into the general payload monitoring SW would be much easier to use.

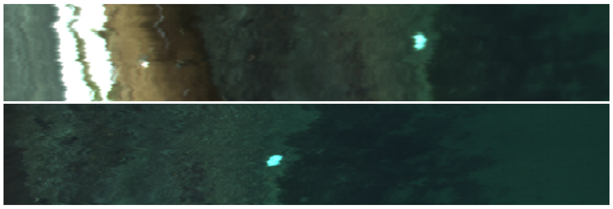


Fig. 2. There are two reflectance targets visable in the top image. One on land and one underwater. The reflectance target to the right in the top image was not covered in a follow-up flight (bottom image), due to improper planning or lack of re-planning.

2.6. Case where it benefited

A field experiment was performed, where a new model of the Sigernes HSIv4 [11] was tested on a multicopter drone, and simultaneously collecting hyperspectral images of shallow waters to learn more about the local shallow water ecosystem and how to monitor them. The simplified system architecture for the field test is shown in Fig. 3, except for this mission, the waterfall visualization on ground, was not ready.

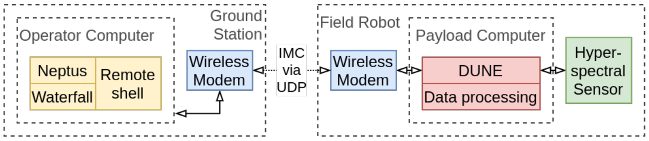
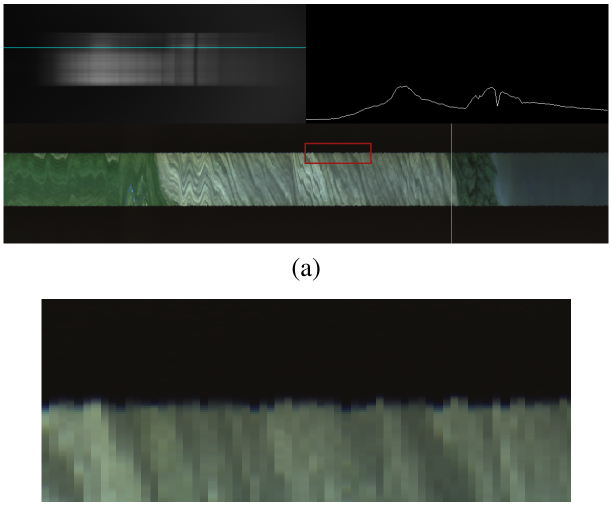


Fig. 3. System setup used for real time hyperspectral data monitoring following the proposed method.

After the first two flights, while the batteries of the multicopter platform were being replaced, the data was inspected with the tool. Anomalies in the data were noticed that can be described as small random spatial shifts in every line, Fig. 4. These indicated that an optical component had come loose. An older model of the hyperspectral camera that did not have the same defect was used to replace the new one with the loose component in subsequent flights.

The real time intensity statistics have been used during multiple field tests to set the exposure time that ensures good signal to noise ratio.



(a)

(b)

Fig. 4. (a) data cube viewed using the proposed tool. In the version of the camera used, not all of the image sensor area was exposed to light, hence the black bars above and below. The area shown in (b) is highlighted by the red rectangle in (a) and shows that the path of light onto the sensor was changing slightly frame to frame, indicating loose optical components.

3. CONCLUSION

While the individual procedures to visualize hyperspectral data presented in this paper are not novel, integrating them into an open source framework like the LSTS toolchain [14] to enable real time mission re-planning has been valuable to the field test operations of the authors.

The proposed open SW tools have proven their worth for quick and easy inspection of the data in the field, to verify that an operation is going as planned, and to verify desired coverage or signal level.

For more in-depth analysis in the lab, tools like ENVI, MATLAB or python, possibly in the form of Jupyter notebooks will be more appropriate.

4. REFERENCES

[1] “AFX10,” VIS-NIR hyperspectral camera, Specim, Spectral Imaging Ltd, https://www.specim.fi/afx.

[2] M. Pepe, L. Fregonese, and M. Scaioni, “Planning airborne photogrammetry and remote-sensing missions with modern platforms and sensors,” *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 412–436, 2018.

[3] M. Cui, A. Razdan, J. Hu, and P. Wonka, “Interactive hyperspectral image visualization using convex optimization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 6, pp. 1673–1684, June 2009.

[4] D. Liu, L. Wang, and J. A. Benediktsson, "An interactive color visualization method with multi-image fusion for hyperspectral imagery," in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2015, pp. 1088–1091.

[5] F. Zhu, Y. Pan, et al., "Interactive visualization of hyperspectral images based on neural networks," *IEEE Computer Graphics and Applications*, vol. 41, no. 5, pp. 57–66, Sep. 2021.

[6] X. Kang, P. Duan, S. Li, and J. A. Benediktsson, "Decolorization-based hyperspectral image visualization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4346–4360, Aug 2018.

[7] H. Su, Q. Du, and P. Du, "Hyperspectral imagery visualization using band selection," in *2012 4th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, June 2012, pp. 1–4.

[8] A. Zare and P. Gader, "Hyperspectral band selection and endmember detection using sparsity promoting priors," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 2, pp. 256–260, April 2008.

[9] S. Ertürk, S. Süer, and H. Koç, "A high-dynamic-range-based approach for the display of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 11, pp. 2001–2004, Nov 2014.

[10] M. D. Crowley, W. Chen, et al., "Visualization of remote hyperspectral image data using google earth," in *2006 IEEE International Symposium on Geoscience and Remote Sensing*, July 2006, pp. 907–910.

[11] F. Sigernes, M. Syrjäsuo, et al., "Do it yourself hyperspectral imager for handheld to airborne operations," *Opt. Express*, vol. 26, no. 5, pp. 6021–6035, Mar 2018.

[12] M. B. Henriksen, E. F. Prentice, et al., "Do-it-yourself vis/nir pushbroom hyperspectral imager with c-mount optics," *Opt. Continuum*, vol. 1, no. 2, pp. 427–441, Feb 2022.

[13] Y. Mao, C. H. Betters, et al., "Openhsi: A complete open-source hyperspectral imaging solution for everyone," *Remote Sensing*, vol. 14, no. 9, 2022.

[14] J. Pinto, P. S. Dias, et al., "The LSTS toolchain for networked vehicle systems," in *2013 MTS/IEEE OCEANS - Bergen*, 2013, pp. 1–9.

[15] "Fast HSI visualization," [Online]. Available: https://github.com/NTNU-SmallSat-Lab/hsi-visualizer.

[16] S. Dawkins, G. Montenegro, et al., "End-to-end performance implications of links with errors," August 2001, [Online]. Available: https://www.rfc-editor.org/rfc/rfc3155.html.

[17] "OpenGL," Low-level graphics API, The Khronos Group Inc., https://www.opengl.org/.

[18] "Simple Fast Multimedia Library," Laurent Gomila, https://www.sfml-dev.org/.

[19] "LodePNG," PNG image decoder and encoder library, Lode Vandevenne, https://lodev.org/lodepng/.