Article

Towards the Concurrent Execution of Multiple Hyperspectral Imaging Applications by Means of Computationally Simple Operations

María Díaz *, **Raúl Guerra** , **Pablo Horstrand** , **Sebastián López** , **José F. López** and **Roberto Sarmiento**

Institute for Applied Microelectronics (IUMA), University of Las Palmas de Gran Canaria (ULPGC), 35001 Las Palmas de Gran Canaria, Las Palmas, Spain; rguerra@iuma.ulpgc.es (R.G.); phorstrand@iuma.ulpgc.es (P.H.); seblopez@iuma.ulpgc.es (S.L.); lopez@iuma.ulpgc.es (J.F.L.); roberto@iuma.ulpgc.es (R.S.)
* Correspondence: mdmartin@iuma.ulpgc.es

check for updates

Abstract: The on-board processing of remotely sensed hyperspectral images is gaining momentum for applications that demand a quick response as an alternative to conventional approaches where the acquired images are off-line processed once they have been transmitted to the ground segment. However, the adoption of this on-board processing strategy brings further challenges for the remote-sensing research community due to the high data rate of the new-generation hyperspectral sensors and the limited amount of available on-board computational resources. This situation becomes even more stringent when different time-sensitive applications coexist, since different tasks must be sequentially processed onto the same computing device. In this work, we have dealt with this issue through the definition of a set of core operations that extracts spectral features useful for many hyperspectral analysis techniques, such as unmixing, compression and target/anomaly detection. Accordingly, it permits the concurrent execution of such techniques reusing operations and thereby requiring much less computational resources than if they were separately executed. In particular, in this manuscript we have verified the goodness of our proposal for the concurrent execution of both the lossy compression and anomaly detection processes in hyperspectral images. To evaluate the performance, several images taken by an unmanned aerial vehicle have been used. The obtained results clearly support the benefits of our proposal not only in terms of accuracy but also in terms of computational burden, achieving a reduction of roughly 50% fewer operations to be executed. Future research lines are focused on extending this methodology to other fields such as target detection, classification and dimensionality reduction.

Keywords: hyperspectral imaging; lossy compression; anomaly detection; hardware-friendly; advanced processing algorithms; concurrent computing; push-broom scanners; line-by-line processing; orthogonal projections; real-time applications

1. Introduction

In recent decades, hyperspectral imagery has experienced a growing popularity, becoming one of the most powerful tools for the Earth observation. In fact, hyperspectral technology has been already used in multiple remote-sensing applications, such as precision agriculture, geoscience, environmental monitoring, urban surveillance and homeland security, among others [1–6], although the continuous evolution of this technology is opening up new ground in potential areas from commercial and industrial applications to biomedicine [7,8]. This rising popularity is rooted in the

great wealth of spectral information that hyperspectral images (HSIs) collect along the electromagnetic spectrum. This permits having a characteristic spectral signature for each image pixel that allows the identification and detection of specific materials and objects in the scene, as well as the estimation of physical parameters of many complex surfaces.

However, hyperspectral image processing poses several challenges due mainly to the high volume of data that must be managed, which typically comprises several Gigabytes per flight [9]. The problem is further complicated by the incorporation of the latest-generation sensors that can produce continual or nearly continual streams of higher-dimensional data [10]. In this context, on-Earth processing has been the mainstream solution for remote-sensing applications that use hyperspectral images. Traditionally, images taken by Earth observation (EO) platforms aboard satellites or manned/unmanned aerial vehicles are downloaded to the ground segment where they are off-line processed on supercomputing systems typically based on Graphics Processing Units (GPUs), Central Processing Units (CPUs), heterogeneous CPU/GPU architectures, or even Field-Programmable Gate Array (FPGAs) [11]. This has been done in this way in order to reduce the computational burden of processes that are executed on-board due to the limitations in the available on-board computational resources as well as the restrictions in power budget and storage space [12,13]. In the space domain, FPGAs have consolidated as the standard choice for on-board remote-sensing processing due to their smaller size, weight and power consumption, as well as for the existence of radiation-hardened and radiation-tolerant FPGAs [14]. However, these devices are more expensive, physically larger and are often technology generations behind in both performance and functionality than their commercial counterparts [14,15]. Regarding GPUs, they have evolved into a highly parallel, multithreaded, many-core processors with tremendous computational speed and very high memory bandwidth [16]. However, they exhibit very high power dissipation figures and are not radiance-tolerant, which has prevented their full incorporation to spaceborne Earth observation missions [16]. Fortunately, the emergence of computing boards that embed low energy consumption GPUs has made more attractive their use, especially in on-board applications carried out by unmanned aerial vehicles (UAVs) [9,17,18]. Nevertheless, these low-power GPUs (LPGPUs) are not as high performing as the same generation desktop GPUs [19,20].

Unfortunately, the data transmission from the remote-sensing platforms to the Earth surface introduces important delays related to the communication of a large amount of data between the source and the final target, producing a bottleneck that can seriously reduce the effectiveness of real-time applications or applications that demand prompt replies [21,22]. Consequently, real-time on-board processing has become a very interesting topic within the remote-sensing field to provide a solution to this type of applications [14,15,22,23]. In the space domain, this is due to the fact that both the acquisition rates of the next-generation hyperspectral sensors and the computational capabilities of the latest-generation space-grade hardware devices are growing [24]. Additionally, the trend for CubeSats is to use common System on Chip (SoC) devices with commercial FPGAs due to their superior performance in terms of power, speed and resources compared to radiation-hardened FPGAs [15,25]. On the contrary, the transmission bandwidth of the data link has been kept relatively stable [26,27]. Accordingly, for being able to transfer to the Earth surface all the acquired data, it is necessary to achieve much higher compression ratios, moving from lossless to lossy hyperspectral compression [28–30]. The alternative is to analyze on-board the hyperspectral data in order to transmit just the obtained results or discard the information that is not of interest for the targeted applications, which reduces the total amount of data to be sent to the Earth surface as well as the required compression ratios. Regarding manned/unmanned aerial vehicles, the interest of on-board carrying out the hyperspectral data analysis is more related to the necessity of obtaining real-time results as well as reducing the storage requirements [9,31].

Nevertheless, in order to carry out this second alternative and execute on-board the hyperspectral data processing, it is still necessary to fulfill many requirements imposed by the available aboard hardware devices and to have in consideration that the computational resources are limited.

Regrettably, the algorithms traditionally proposed for hyperspectral analysis have been addressed as independent entities, using those mathematical methods that better maximize the results for each particular case. In addition, these approaches normally give rise to complex algorithms with many data dependencies. This is because the algorithm development phase is usually detached from the hardware implementation stage, resulting in very inefficient hardware implementations. All of this makes the on-board execution of multiple processes for analyzing the acquired hyperspectral data not fully viable, especially under real-time constraints [6,14]. Additionally, in view of the computationally intensive nature of state-of-the-art hyperspectral imaging algorithms, new hardware-friendly solutions are required enabling real-time execution based on an appropriate trade-off among design requirements [14].

Considering all these facts, in this manuscript we have dealt with the issue around the on-board execution of multiple hyperspectral analysis techniques onto the same hardware device and concurrently, understanding it as simultaneously. On this basis, a new algorithmic solution is proposed in this work, based on a set of common core operations that allows the concurrent execution of multiple hyperspectral analysis processes at a reasonable computational burden. As many other state-of-the-art approaches for analyzing hyperspectral images [32–38], the proposed set of core operations is based on orthogonal projection techniques. Concretely, the proposed method uses a modified version of the Gram–Schmidt orthogonalization process [39,40] that allows extraction of useful information for many different hyperspectral imaging applications while at the same time, ensuring a low computational complexity and a high level of parallelism. The advantages of this set of core operations have already been tested using different algorithms specifically developed for unmixing applications [40], hyperspectral lossy compression [41] and anomaly detection [42,43]. Additionally, some of these algorithms have been successfully implemented into parallel computing devices, such as LPGPUs mounted on a UAV [20] and next-generation space-grade FPGAs [44], achieving real-time performance in both cases.

Concretely, in this work the proposed set of core operations has been optimized in such a way that it can be executed just once and reuse the obtained results for carrying out multiple hyperspectral analysis in a concurrent manner, thus reducing the overall computational cost and the required hardware resources. In particular, we have focused on the lossy compression of HSIs and the detection of anomalous pixels to demonstrate the goodness of the proposed methodology. The reasons why we have dealt with the compression issue have been clearly analyzed in above lines. In relation with anomaly detection, we have decided to focus on applications in which a prompt response is mandatory [45,46], such as the one introduced in [47] within the European H2020 project ENABLE-S3 (European Initiative to Enable Validation for Highly Automated Safe and Secure Systems). The goal of this work was the automated driving of a harvester for wildlife friendly agro-operation. To do this, a push-broom hyperspectral camera mounted on a UAV inspects the terrain to find big obstacles (animals or rocks) to prevent fatal accidents due to agricultural mowing operations. In this scenario, anomaly detection can be used to find these obstacles. Due to inherent nature of this application, the on-board processing of the captured images is mandatory, leaving no room for the off-line processing of them. Nevertheless, we would like to mention that this methodology could be potentially extended to include other processes without a relevant increment of the required computational resources and without affecting the quality of the obtained results.

The rest of this paper is organized as follows. Section 2 deepens in the proposed set of core operations and provides a detailed description of the proposed methodology for the concurrent execution of lossy compression for HSIs and anomaly detection. Section 3 describes the hyperspectral data sets, the performance assessment metrics used to evaluate the accuracy of the results provided by the proposed solution and also shows the obtained experimental results. In Section 4, we discuss the strengths and limitations of our method as well as its potential to be extended to other remote-sensing applications. Finally, Section 5 collects the drawn conclusions.

2. Materials and Methods

2.1. Proposed Set of Core Operations

Hyperspectral imaging gathers spectral information of hundreds of continuous and narrow wavelengths along the electromagnetic spectrum. Therefore, any single pixel within an HSI is associated with a full continuous spectrum in a given range, commonly called spectral signature of the pixel. HSIs are spectrally and spatially smooth which means that nearby spectra and wavelengths are highly correlated [16]. For this reason, pixels within an HSI can be grouped according to their spectral similarities and represented as a combination of relatively few spectral signatures that are representative of each cluster. As a consequence, features of HSIs normally reside in a subspace that normally has a much smaller dimension than the original number of spectral bands.

On this basis, the extraction of the most representative pixels of materials present in a scene permits performing dimensionality reduction and thus, to compress the HSIs as analyzed in [41] for the lossy compression of HSIs. Considering that a HSI is composed of pe pixels with nb spectral bands, if these pe pixels are projected onto a subset of the p most different pixels within the scene, the original pe pixels can be represented as a linear combination of their projections onto those p pixels. Therefore, the original hyperspectral cube can be represented on a new subspace of dimension p, being $p << nb$, and hence, to be compressed. The number of p selected pixels directly determines the compression ratio achieved in the compression process.

In addition, this methodology eases to perform other hyperspectral imaging analysis techniques, such as unmixing, target detection and anomaly detection. In the linear mixing model, the spectrum of a mixed pixel is a linear combination of the endmember spectra weighted by the fractional area coverage by each endmember in a pixel, commonly named abundance. In this case, endmembers can be seen as the p most different pixels in an image while abundances can be derived from the projection of each image pixel onto each endmember [40]. In the particular case of anomaly detection, this set of p vectors can be employed to estimate the subspace spanned by the background samples in which the projection of the anomalous pixels are notoriously lower [43]. Similarly, in the field of target detection this methodology can be used to extract the most representative pixels of the background, also commonly referred to as undesired signatures, to be later used to annihilate from image pixels the spectral information that does not belong to the desired target [48].

The main conclusion to be drawn from the above is that many hyperspectral imagery processing techniques may be performed using the same mathematical methods. However, data collected by the hyperspectral sensors are currently processed using different kinds of mathematical algorithms to extract the information that is useful for the targeted hyperspectral imaging application. Furthermore, the complexity of the state-of-the-art algorithms together with the amount of data to be processed normally result in very high computational times, making the hyperspectral technology not useful for applications under tight latency/power/memory constraints.

With this in mind, we have gone a step further in this work through the definition of a set of common core operations that extract features from the HSIs useful for many applications. As a novelty, it permits the concurrent execution of many different tasks at the same time with the advantage of sharing the most computationally intensive operations. To do this, our proposal is based on orthogonal projection techniques and, more specifically, on a modified version of the well-known Gram—Schmidt orthogonalization method [39,40]. This modified version of the Gram—Schmidt method features low computational complexity since non-complex matrix calculation is involved and previously computed information is reused.

Additionally, the proposed set of core operations can be efficiently and independently applied on blocks of image pixels without requiring any specific spatial alignment. This feature makes our proposal a promising solution for real-time applications, especially when using hyperspectral sensors based on push-broom/whisk-broom scanners since lines of image pixels, also named hyperspectral frames, can be processed as soon as they are sensed. In addition, the possibility of independently

processing blocks of pixels as soon as they are sensed avoids storing and processing large amount of data, thereby reducing the amount of required hardware resources and also speeding up the entire process.

In the remaining of this section, we deeply analyze the proposed set of common core operations as well as the Gram–Schmidt method to ease the understanding of the proposed methodology.

2.1.1. General Notations

In the following, $\mathbf{HI} = \{\mathbf{F}_i, i = 1, ..., nr\}$ is a sequence of nr hyperspectral frames or lines of pixels, $\mathbf{F}_i$, comprised by nc pixels with nb spectral bands. Pixels within $\mathbf{HI}$ are grouped in blocks of BS pixels, $\mathbf{M}_k = \{\mathbf{r}_j, j = 1, ..., BS\}$, being normally BS equal to nc, or multiple of it, and k spans from 1 to $\frac{nr \cdot nc}{BS}$. $\mathbf{X} = \{\mathbf{x}_j, j = 1, ..., BS\}$ is an auxiliary copy of each image block $\mathbf{M}_k$. $\mathbf{E} = \{\mathbf{e}_n, n = 1, ..., p\}$ saves the p most different hyperspectral pixels extracted from each $\mathbf{M}_k$ block. $\mathbf{V} = \{\mathbf{v}_n, n = 1, ..., p\}$ comprises p vectors of BS elements where each $\mathbf{v}_n$ vector corresponds to the projection of the BS pixels within $\mathbf{M}_k$ onto the corresponding n extracted pixel, $\mathbf{e}_n$. $\mathbf{Q} = \{\mathbf{q}_n, n = 1, ..., p\}$ and $\mathbf{U} = \{\mathbf{u}_n, n = 1, ..., p\}$ save p pixels of nb bands that are orthogonal among them. Finally, $\mathbf{P}$ is the orthogonal projection matrix that is a square matrix of dimension $nb \cdot nb$ and $\mathbf{I}$ is the identity matrix with the same dimensions.

2.1.2. Gram-Schmidt Method

The Gram–Schmidt method calculates the orthogonal projection of a vector $\mathbf{e}_i$ to a set of vectors $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_j]$, with $j < i$, by subtracting the portion of the vector $\mathbf{e}_i$ contained in the directions spanned by the vectors $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_j]$. Consequently, the Gram–Schmidt method allows orthogonalization of a set of independent vectors $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p]$ and brings as a result a set of orthogonal vectors $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_p]$ and their normalized vectors $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p]$.

In this work, we employed a modified version of the Gram–Schmidt method where vectors $\mathbf{u}_p$ are normalized dividing by the squared of its $l^2 norm$. This modified version of the Gram–Schmidt method features low computational complexity since simple matrix operations are involved and, allows the reuse of previously computed information, which is reflected in speeding up the overall process.

The pseudocode of the modified version of the Gram–Schmidt method is shown in Algorithm 1, where "′" represents the transpose of a vector. It has to be mentioned that the transpose of a vector, $\mathbf{x}'$, followed by "." and other vector, $\mathbf{y}$, means the dot product between these two vectors. Similarly, a vector, $\mathbf{x}'$, followed by "." and a set of vectors, $\mathbf{Y}$, means the dot product between $\mathbf{x}'$ and each column vector within $\mathbf{Y}$. This naming has been adopted throughout the manuscript. As it can be seen, the first orthogonal vector within $\mathbf{Q}$ is $\mathbf{e}_1$. In the second iteration, $n = 2$, $\mathbf{e}_2$ is projected on the direction spanned by $\mathbf{q}_1 = \mathbf{e}_1$, giving the projection vector $\mathbf{v}_1$ in Line 4. Then, it is subtracted from $\mathbf{q}_2 = \mathbf{e}_2$ in Line 5. As a result, $\mathbf{q}_2$ contains the spectral information spanned by $\mathbf{e}_2$ that cannot be represented by $\mathbf{e}_1$, i.e., orthogonal to $\mathbf{e}_1$. The process is repeated until the p vectors are orthogonal to each other.

Algorithm 1 Modified version of the Gram–Schmidt Orthogonalization

---

    Inputs:
    $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p]$
    Outputs:
    $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_p]$ {Orthogonalized vector}; $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p]$ {Orthonormalized vector};
    Algorithm:
 1: **for** $n = 1$ **to** p **do**
 2:     $\mathbf{q}_n = \mathbf{e}_n$
 3:     **for** $j = 1$ **to** $n - 1$ **do**
 4:         $\mathbf{v}_j = \mathbf{u}'_j \cdot \mathbf{e}_n$;
 5:         $\mathbf{q}_n = \mathbf{q}_n - \mathbf{q}_n \cdot \mathbf{v}_j$;
 6:     **end for**
 7:     $\mathbf{u}_n = \mathbf{q}_n / (\mathbf{q}'_n \cdot \mathbf{q}_n)$;
 8: **end for**

2.1.3. Set of Core Operations

As already mentioned, the proposed set of core operations allows performing the simultaneous extraction of the p most representative hyperspectral pixels in a scene and identifying the amount of the image spectral information that can be represented by them. To do this, our proposal is based on the modified version of the Gram–Schmidt method that was further analyzed in Section 2.1.2. The first extracted representative pixel ($\mathbf{e}_1$), also called characteristic pixels in the remaining of this manuscript, is the pixel of the HSI with the highest deviation from the average pixel or centroid pixel $\hat{\mu}$. Afterward, the orthogonal projection of each image pixel with respect to $\mathbf{e}_1$ is performed using the aforementioned Gram–Schmidt method. At this point, image pixels just retain the information that is not contained by $\mathbf{e}_1$ and thus, that is orthogonal to it. Once the first endmember has been selected, the proposed set of core operations sequentially extracts new characteristic pixels by selecting the pixels with the largest orthogonal projections to the pixels already extracted. With it, we achieve to select the most different pixels in each iteration, understanding as it, those pixels that cannot be well represented by previously selected pixels.

The proposed set of core operations is uncovered in Algorithm 2 for an image block, $\mathbf{M}_k$. Operations from 4 to 14 are repeated p times to extract the p most different pixels. First of all, the hyperspectral image block, $\mathbf{M}_k$, is centered in Line 2, obtaining matrix $\mathbf{C}$ by subtracting the centroid or average pixel, $\hat{\mu}$, to all pixels within $\mathbf{M}_k$. An auxiliary copy of $\mathbf{C}$ is stored in matrix $\mathbf{X}$. Secondly, the pixels are sequentially extracted from lines 4 to 13. In this process, the dot product of each frame pixel within $\mathbf{X}$ with itself is first calculated from lines 5 to 7. In the remainder of this document, it is referred as brightness of a pixel. The extracted pixels $\mathbf{e}_n$ are selected as those pixels from $\mathbf{M}_k$ that correspond to the highest brightness in matrix $\mathbf{X}$, as shown in line 9. Then, the orthogonal projection vectors $\mathbf{q}_n$ and $\mathbf{u}_n$ are accordingly obtained as shown in lines 10 and 11, respectively. After that, the information that can be spanned by the defined $\mathbf{q}_n$ and $\mathbf{u}_n$ orthogonal vectors is stored in the projected image vector $\mathbf{v}_n$ and subtracted to $\mathbf{X}$ in lines 12 and 13. As it can be seen, Lines 2 and 7 of Algorithm 1 corresponds to Lines 10 and 11 of Algorithm 2, respectively. Similarly, operations shown in Lines 4 and 5 of Algorithm 1 corresponds to Lines 12 and 13 of Algorithm 2.

Algorithm 2 Set of core operations

    Inputs:
    $\mathbf{M}_k = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{BS}]$
    Outputs:
    $\hat{\mu}$ {Average Pixel}; $\mathbf{E} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_p]$ {Characteristic pixels}; $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_p]$ {Orthogonalized

    vectors}; $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_p]$ {Orthonormalized vectors}; $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_p]$ {Projection vectors}
    Algorithm:
  1: Centroid or average pixel: $\hat{\mu}$;
  2: Centralization: $\mathbf{C} = \mathbf{M}_k - \hat{\mu}$;
  3: Auxiliary Copy: $\mathbf{X} = \mathbf{C}$;
  4: **for** $n = 1$ **to** p **do**
  5:     **for** $j = 1$ **to** BS **do**
  6:        Brightness Calculation: $\mathbf{b}_j = \mathbf{x}'_j \cdot \mathbf{x}_j$;
  7:     **end for**
  8:     Maximum Brightness: $j_{max} = \mathrm{argmax}(\mathbf{b}_j)$;
  9:     Extracted pixels: $\mathbf{e}_n = \mathbf{r}_{j_{max}}$;
10:     $\mathbf{q}_n = \mathbf{x}_{j_{max}}$;
11:     $\mathbf{u}_n = \mathbf{q}_n / b_{j_{max}}$;
12:     Projection vector: $\mathbf{v}_n = \mathbf{u}'_n \cdot \mathbf{X}$;
13:     Information Subtraction: $\mathbf{X} = \mathbf{X} - \mathbf{q}_n \cdot \mathbf{v}_n$;
14: **end for**

### 2.1.4. Discussion about the Proposed Set of Core Operations

From analysis made in the previous Section 2.1.3, it can be concluded that a HSI can be represented as a function of some image pixels, $\mathbf{E}$, and their corresponding projection vectors $\mathbf{V}$. For this reason, if a HSI is compound of pe pixels, being $pe = nr \cdot nc$, and nb spectral bands, it can be now represented by $p \cdot (nb + pe)$ elements, being $p <<< nb$, and then, getting the original image compressed. As it can be seen, the compression ratio directly depends on the number of selected pixels, p. Therefore, if less p vectors are extracted, more compression is obtained but on the contrary, more information is lost during the compression-decompression process. It is because image $\mathbf{X}$ defined in Algorithm 2 retains the spectral information of the image that is not contained in the already extracted characteristic pixels. Therefore, smaller the number of extracted pixels, bigger the information that cannot be represented by them. For this reason, the proposed set of core operations acts as an spectral transform where image pixels are projected onto a new subspace spanned by the p selected pixels and thus, it can be used to perform the lossy compression of HSI.

In the field of anomaly detection, the anomalous pixels are spectral signatures that significantly differ from the background distribution and hence, cannot be well represented by it. For this reason, the projection of the anomalous pixels is significantly higher in the orthogonal subspace to the one spanned by the background samples. On this basis, the solution for the anomaly detection issue lies in modeling the whole background and subtracting it from every image pixel by means of orthogonal subspace projections. However, the mixing model, the p most characteristic pixels of a subsequent set of hyperspectral frames, $\mathbf{M}_k$, can be used to represent the background distribution. In addition, the projection separation statistic for an image pixel $\mathbf{r}_j$ can be calculated using the orthogonal projection matrix, $\mathbf{P} = \mathbf{I} - \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$, where $\mathbf{W} = [\mathbf{w}_n, n = 1, ..., p]$ is a matrix whose columns are the p projection basis obtained from the background samples [49]. Therefore, the proposed set of core operations can be employed to extract the p most characteristic pixels of the background and to estimate the orthogonal projection matrix, $\mathbf{P}$.

2.2. Proposed Methodology for Lossy Compression and Anomaly Detection

From the analysis made in Sections 2.1.3 and 2.1.4, we can conclude that a joint solution that permits the on-board performance of both anomaly detection and lossy compression of HSIs simultaneously and in real time is feasible. In this Section, we explain in detail how the outputs of the proposed set of core operations can be reused to perform both processes concurrently, using much less computational resources than if both of them were separately executed. Figure 1 shows the flow-chart followed by our proposal to perform both the compression and the anomaly detection processes for a particular image block, $\mathbf{M}_k$.
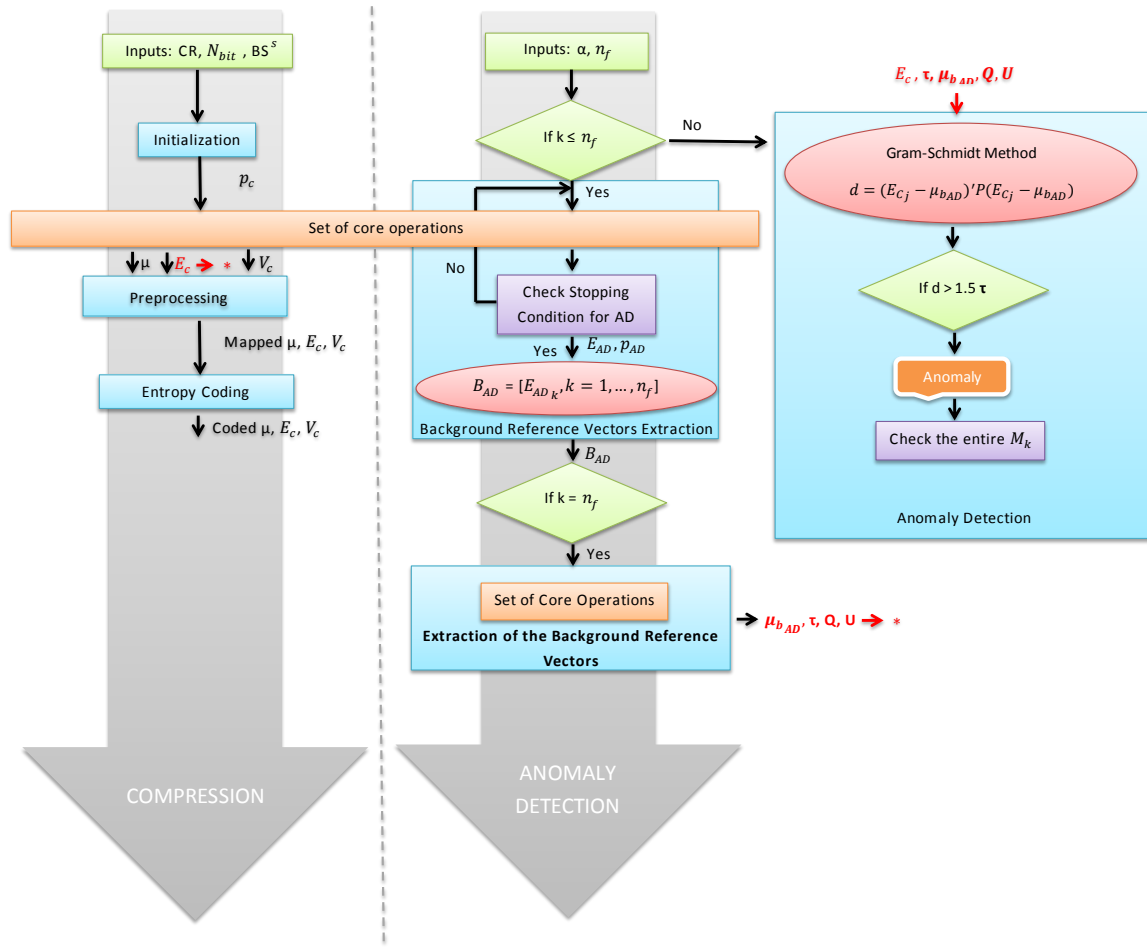
Figure 1. FLow-chart of the proposed algorithm stages for each image block $\mathbf{M}_k$. $\rightarrow$ * means that these variables are reused in other stages of the algorithm. If $p_c > p_{AD}$, then E_{AD} is a subset of E_c and vice-versa.

2.2.1. Set of Core Operations for Lossy Compression and Anomaly Detection

As analyzed in Section 2.1.4, the lossy compression of HSIs as well as the detection of anomalous pixels can be addressed using the same mathematical method and, in particular, by means of the set of core operations proposed in this work. In the field of hyperspectral lossy compression, the most different pixels, $\mathbf{E}$, and their projection vectors, $\mathbf{V}$, extracted from each image block, $\mathbf{M}_k$, can be used to decorrelate the images and thereby, to compress them. In the field of anomaly detection, the most characteristic pixels, $\mathbf{E}$, within an image block, $\mathbf{M}_k$, and their corresponding sets of orthogonal vectors, $\mathbf{Q}$ and $\mathbf{U}$, can be used to estimate the orthogonal subspace spanned by the background distribution where anomalous pixels can be better represented. On this basis, the first step to be followed consists

of extracting the most representative pixels within each image block, $\mathbf{M}_k$, using the proposed set of core operations. However, this number of pixels could be different for each process to be performed.

Regarding the compression process, the number of the most different pixels, $\mathbf{E}$, and projection vectors, $\mathbf{V}$, extracted from each image block, $\mathbf{M}_k$, directly depends on the compression ratio achieved by our method. This number will be referred as to p_c and estimated following Equation (1). In this equation, CR represents the minimum compression ratio to be reached for the targeted application, DR refers to the number of bits employed to represent each element of $\mathbf{M}_k$ and $\mathbf{E}$, and N_{bits} determines the number of bits used for representing the values of $\mathbf{V}$. This preliminary stage where p_c is computed corresponds with the blue square named *Initialization* in Figure 1. Once p_c is set, the core operations defined in Section 2.1.3 and in Algorithm 2 are performed in order to extract the p_c most different pixels, $\mathbf{E}$, and their corresponding projection vectors, $\mathbf{V}$. They are referred to as $\mathbf{E_c}$ and $\mathbf{V_c}$, respectively, in the remainder of this document. These sets of vectors, jointly with the average pixel, $\boldsymbol{\mu}$, of the image block, $\mathbf{M}_k$, under analysis are the outputs needed by the subsequent algorithm stages for the compression process, as it will be further explained in Section 2.2.2.

$$p \leq \frac{DR \cdot nb \cdot (BS - CR) - CR}{CR \cdot (DR \cdot nb + N_{\text{bits}} \cdot BS)} \tag{1}$$

Regarding the anomaly detection process, the extraction of the most characteristic pixels for modeling the background distribution, $\mathbf{E}$, must be done at first place. To do this, our proposal independently processes the first sensed n_f hyperspectral image block, $\mathbf{M}_k$, under the assumption that they are free of anomalous signatures and hence, fully representative of the background distribution, although background samples obtained in previous flights may be also used instead of extracting them from the first n_f frames. Nevertheless, the number of extracted pixels for each image block, $\mathbf{M}_k$, cannot be enough if we use p_c since it is estimated as a function of the desired compression ratio, CR. Bigger the desired CR, fewer pixels can be transmitted and hence, worst the background is represented. In this case, we use the remaining information that cannot be represented by the extracted $\mathbf{e}_n$ vectors, as it was explained in Section 2.1.4, to estimate a good number of pixels representative of the background for each image block, $\mathbf{M}_k$. On this basis, if no more $\mathbf{e}_n$ pixels are selected, it means that a small part of the spectral information is lost when the image is reconstructed using the p selected $\mathbf{e}_n$ pixels. This loss can be measured through the maximum brightness, $b_{j_{\max}}$, of $\mathbf{X}$ after the last pixel is selected. The process finishes when the loss, in percentage terms, is less than α, as it can be seen in Equation (2), where $\mathbf{c}'_{j_{\max}} \cdot \mathbf{c}_{j_{\max}}$ represents the dot product of the pixel $\mathbf{r}_j$ with itself in the centralized version of the original image, $\mathbf{C}$. Accordingly, α represents the percentage of the information that will be considered to be noise. In general, experience has shown that a stop factor, α, fixed to 1% is sufficient. As it can be seen, the number of extracted reference vectors, p, can be different for each image block, $\mathbf{M}_k$. In order to distinguish p and $\mathbf{E}$ from those needed by the compression process, they are referred to as p_{AD} and $\mathbf{E_{AD}}$, respectively,

$$\frac{b_{j_{\max}}}{\mathbf{c}'_{j_{\max}} \cdot \mathbf{c}_{j_{\max}}} \cdot 100 < \alpha \rightarrow \text{Stop selecting } p_{AD} \ e_n \text{ pixels} \tag{2}$$

Finally, since both the compression and anomaly detection processes need to perform the set of core operations to extract their associated vectors, it is executed just once, and its outputs are reused by both processes. Nevertheless, the number of p iterations in Algorithm 2 are determined by the requirements imposed by p_c and p_{AD} in such a way that if $p_{AD} < p_c$, then $\mathbf{E_{AD}}$ is a subset of $\mathbf{E_c}$. Otherwise, $\mathbf{E_c}$ is a subset of $\mathbf{E_{AD}}$.

2.2.2. Subsequent Stages for Lossy Compression

After the extraction of the p_c characteristic pixels, $\mathbf{e_{c_n}}$, and projection vectors, $\mathbf{v_{c_n}}$, using the set of core operations, our proposal follows the same methodology as [41] to slightly increase the compression ratio at a very low computational cost and without introducing further losses of information. To do

this, outputs $\mathbf{E_c}$, $\mathbf{V_c}$ and μ are independently processed in two stages, named *Preprocessing* and *Entropy Coding* in Figure 1.

The main goal of the *Preprocessing* stage is to transform the aforementioned outputs to have only positive integer elements closer to zero using the same dynamic range as the original ones. It is done to ease the subsequent entropy coding stage that takes advantage of the redundancies within the data to assign the shortest word length to the most common values. To perform the aforementioned transformation, the prediction error mapper described in the Consultative Committee for Space Data Systems (CCSDS) recommended standard for lossless multispectral and hyperspectral image compression [50] is used. Particularly for $\mathbf{V_c}$, it is also scaled to positive integer values before the error mapping to fully exploit the available dynamic range according to the input parameter N_{bits}. This is because $\mathbf{V_c}$ contains the projection of the image pixels, $\mathbf{x}_j$, into the space spanned by the different orthogonal vectors, $\mathbf{u}_n$, extracted in each iteration of the set of core operations. Hence, the value range of its elements, $\mathbf{v_c}$, is between $(-1,1]$. Finally, *Entropy Coding* is the last stage of the compression process. In this stage, each single output vector μ, $\mathbf{e}_{\mathbf{c}_n}$ and $\mathbf{v}_{\mathbf{c}_n}$ is independently coded using a Golomb–Rice coder [51]. For more information, we encourage the reader to see [41].

2.2.3. Subsequent Stages for Anomaly Detection

As mentioned in Section 2.2.1, different $\mathbf{E_{AD}}$ are extracted each time that an image block, $\mathbf{M}_k$, is processed. Additionally, if each set of pixels $\mathbf{E_{AD}}_k$ is saved in a bigger matrix, $\mathbf{B_{AD}}$, a lot of them may represent the same material or entity since image blocks, $\mathbf{M}_k$, are independently processed.

In subspace-based anomaly detectors, anomalous pixels are better detectable in the orthogonal subspace to the one spanned by the background distribution. For this reason, it is necessary to extract the purest background reference vectors within $\mathbf{B_{AD}}$. For doing so, the set of core operations can be applied once again in background, playing $\mathbf{B_{AD}}$ as input matrix $\mathbf{M}_k$. As outputs, we obtain the orthogonal vectors $\mathbf{Q}$ and $\mathbf{U}$, which are later used to compute the projection matrix, $\mathbf{P}$; the average pixel of the background distribution, $\mu_{b\,\mathbf{AD}}$, which is used to centralize the subsequent sensed image block, $\mathbf{M}_k$; and a threshold, $\tau = b_{j_{\max}}$, which can be used as a benchmark to later identify anomalous pixels in Section 2.2.3. This stage is referred as to *Extraction of the Background Reference Vectors* in Figure 1.

From analysis made in above lines, the background statistics are inferred from the first n_f images blocks, $\mathbf{M}_k$. For this reason, the anomaly detection process itself is performed onto the following scanned image blocks, $\mathbf{M}_k$. However, the subspace that is orthogonal to the one spanned by the background samples must be first estimated. In subspace-based anomaly detectors, the projection separation statistic for an image pixel $\mathbf{r_j}$ is calculated as [49]:

$$d = (\mathbf{r_j} - \hat{\mu}_b)^T \cdot \mathbf{P} \cdot (\mathbf{r_j} - \hat{\mu}_b) \tag{3}$$

where $\hat{\mu}_b$ is the estimation of the background sample average pixel, $\mu_{b\,\mathbf{AD}}$ in this manuscript.

Computing the orthogonal subspace matrix $\mathbf{P}$, defined in Section 2.1.4 as $\mathbf{P} = \mathbf{I} - \mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T$, may be a real challenge since it implies matrix inverse whose dimension directly depends on the number of background samples p_{AD}. On this basis, [43] proposes a hardware-friendly solution using outputs $\mathbf{Q}$ and $\mathbf{U}$, previously worked out. Concretely, $\mathbf{Q}$ is equivalent to $\mathbf{W}$ and $\mathbf{U}$ to $\mathbf{W}(\mathbf{W}^T\mathbf{W})^{-1}$. Therefore, the matrix $\mathbf{P}$ can be calculated from $\mathbf{P} = \mathbf{I} - \mathbf{Q}\mathbf{U}^T$.

Nevertheless, d actually represents the information of an image pixel, $\mathbf{r_j}$, that is not contained in the subspace spanned by the background samples, i.e., which is orthogonal to them. As a consequence, operations involved in Equation (3) and in the computation of $\mathbf{P}$ can be replaced by the Gram–Schmidt method described in Section 2.1.2, as it was further analyzed in [52]. For processing each hyperspectral pixel, $\mathbf{r_j}$, using this new alternative, average pixel, $\mu_{b\,\mathbf{AD}}$, is first subtracted from $\mathbf{r_j}$, as shown in Line 2 of Algorithm 3. Afterwards, the information of $\mathbf{r_j}$ that can be spanned by the background samples is also subtracted using $\mathbf{Q}$ and $\mathbf{U}$, as shown in Lines 3 to 6 of Algorithm 3. Finally, the amount of information of $\mathbf{r_j}$ that is orthogonal to the space spanned by the background samples is measured in

line 7 as the dot product of the remainder of $\mathbf{r}_j$ with itself. As it was concluded in [43], parameter τ is already scaled in the orthogonal subspace spanned by P. For this reason, it can be used as a threshold to detect the anomalous pixels. Hence, an anomalous pixel is considered to be a mild outlier whose projection d is larger than $1.5 \cdot \tau$.

Algorithm 3 Alternative method to compute the orthogonal projection matrix

---

    Inputs:
    $\mathbf{M}_k = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{BS}], \mathbf{Q}, \mathbf{U}, \boldsymbol{\mu}_{b\,\mathbf{AD}}, \tau$
    Algorithm:
  1: **for** i $= 1$ **to** BS **do**
  2:      $\mathbf{r}_j = \mathbf{r}_j - \boldsymbol{\mu}_{b\,\mathbf{AD}}$
  3:      **for** k $= 1$ **to** p_{AD} **do**
  4:         $v = \mathbf{U}_k' \cdot \mathbf{r}_j$
  5:         $\mathbf{r}_j = \mathbf{r}_j - \mathbf{Q}_k \cdot v$
  6:      **end for**
  7:      $d_j = \mathbf{r}_j' \cdot \mathbf{r}_1$
  8:      **if** $d_i \leq 1.5 \cdot \tau$ **then**
  9:         Anomalous pixel
10:      **end if**
11: **end for**

Although the p_{AD} background reference vectors were already extracted from the previous n_f image blocks, $\mathbf{M}_k$, the set of core operations described in Section 2.1.3 are still being run on new received hyperspectral frames to extract the p_c most characteristic pixels, $\mathbf{E_c}$, needed for the compression process. On the basis that this $\mathbf{E_c}$ vectors are the most different pixels within $\mathbf{M}_k$, they collect the rarest signatures too. For this reason, if any anomalous pixel is present in $\mathbf{M}_k$, it must be collected in $\mathbf{E_c}$. Hence, just these $\mathbf{E_c}$ vectors are projected onto the space spanned by $\mathbf{P}$. In case of presence of any anomalous pixel within $\mathbf{E_c}$, the entire image block, $\mathbf{M}_k$, is processed in order to also detect mixed anomalous pixels. Otherwise, only $\mathbf{E_c}$ vectors are checked, thus reducing the number of operations to be performed.

2.2.4. Computational Complexity of the Proposed Methodology

In this section, the computational complexity of the proposed methodology is evaluated in terms of the number of floating-point operations (FLOPs) involved in each stage of the algorithm. For clarity, FLOPs are simple calculations such as addition, subtraction, multiplication, and division. On the basis that our proposal has been specifically designed for being able to independently process blocks of image pixels, $\mathbf{M}_k$, different number of FLOPS are executed according to their index, k, as it can be noticed from Figure 1. For this reason, algorithm stages have been redefined in this analysis according to the range of possible values of k, i.e., $k < n_f, k = n_f$ and $k > n_f$. Consequently, Table 1 collects the number of FLOPs required to process one image block, $\mathbf{M}_k$, as a function of k. In this Table 1, *Core operations* collects the number of operations involved by the proposed set of core operations defined in Algorithm 2, *Codification* represents the *Preprocessing* and *Entropy Coding* stages belonging to the compression process and *Gram–Schmidt* collects the number of operations described in Algorithm 3 to compute the projection separation statistic, d.

Table 1. Number of FLOPs and computational complexity of each stage of the proposed methodology.

Stage		FLOPs	Complexity
$k < nf$	Core operations	$p_c \cdot (6 \cdot BS \cdot N_b + N_b) + 2 \cdot N_b \cdot BS$	$O(6 \cdot p_c \cdot BS \cdot N_b)$
	Codification	$5 \cdot p_c \cdot (N_b + BS) + 34 \cdot p_c$	$O(5 \cdot p_c \cdot (N_b + BS))$
$k = nf$	Core operations (AD)	$p_{AD} \cdot (6 \cdot p_c \cdot n_f \cdot N_b + N_b) + 2 \cdot N_b \cdot p_c \cdot n_f$	$O(6 \cdot p_{AD} \cdot p_c \cdot n_f \cdot N_b)$
	Core operations	$p_c \cdot (6 \cdot BS \cdot N_b + N_b) + 2 \cdot N_b \cdot BS$	$O(6 \cdot p_c \cdot BS \cdot N_b)$
	Codification	$5 \cdot p_c \cdot (N_b + BS) + 34 \cdot p_c$	$O(5 \cdot p_c \cdot (N_b + BS))$
$k > nf$	Core operations	$p_c \cdot (6 \cdot BS \cdot N_b + N_b) + 2 \cdot N_b \cdot BS$	$O(6 \cdot p_c \cdot BS \cdot N_b)$
	Codification	$5 \cdot p_c \cdot (N_b + BS) + 34 \cdot p_c$	$O(5 \cdot p_c \cdot (N_b + BS))$
	Gram–Schmidt (non-anomalous)	$4 \cdot p_{AD} \cdot p_c \cdot N_b + 3 \cdot p_c \cdot N_b$	$O(4 \cdot p_{AD} \cdot p_c \cdot N_b))$
	Gram–Schmidt (anomalous)	$4 \cdot p_{AD} \cdot N_b \cdot (p_c + BS) + 3 \cdot N_b \cdot (p_c + BS)$	$O(4 \cdot p_{AD} \cdot N_b \cdot (p_c + BS))$

1. $k < n_f$

   In this case, the set of core operations is used to extract the pc and p_{AD} most different pixels in each received image block, $\mathbf{M}_k$. For simplicity, we consider that p_c is equal to p_{AD} in this analysis. In addition, the *Preprocessing* and *Entropy Coding* stages belonging to the compression process are also executed in order to codify the outputs of the core operations before being sent.

2. $k = n_f$

   On one hand, the set of core operations and the *Preprocessing* and *Entropy Coding* stages are applied to perform the compression of the image block at issue. On the other hand, the extraction of the purest background reference vectors within $\mathbf{B_{AD}}$ is also needed by the anomaly detection process as explained in Section 2.2.3. Concretely, this stage is referred as to *Extraction of the Background Reference Vectors* in Figure 1. For doing so, the set of core operations is used once again but in this case, $\mathbf{B_{AD}}$ plays as input matrix $\mathbf{M}_k$. Whereas p_c pixels were extracted from previous frames, $k \leq n_f$, $\mathbf{B_{AD}}$ is composed of $p_c \cdot n_f$ pixels. For this reason, BS is now replaced by $p_c \cdot n_f$ in Table 1. In addition, the number of extracted pixels is referred as to p_{AD} since they could be different from p_c.

3. $k > nf$

   Once the first n_f hyperspectral frames were processed in order to obtain the background statistic needed by the anomaly detection process, the set of core operations are applied in the next received image blocks, $\mathbf{M}_k$, in order to obtain the p_c reference pixels needed by the compression process. As it was explained in Section 2.2.3, these pixels are also used to detect those image blocks where anomalous pixels are present. For doing this, the Gram–Schmidt method is used to estimate the projection separation statistic, defined in Equation (3), for each pixel $\mathbf{e}_{c_j}$. If no pixels within $\mathbf{E_c}$ are anomalous, other pixels within $\mathbf{M}_k$ are not analyzed. Otherwise, the entire image block, $\mathbf{M}_k$, is processed to also detect mixed anomalous pixels. The number of FLOPs involved in this last step are collected in the last two rows of Table 1. The first one collects the number of FLOPs for those $\mathbf{M}_k$ free of anomalous while the second one represents the opposite case. It should be emphasized that this last situation is very unlikely since anomalous pixels have normally a low presence.

   Therefore, for an image, $\mathbf{HI}$, composed of nr hyperspectral frames, $\mathbf{F}_i$, with nc pixels and with a probability $x \leq 1$ of anomalous $\mathbf{M}_k$, the total number of FLOPs to compute both the lossy compression and anomaly detection processes is $\frac{nr \cdot nc}{BS} \cdot$ *(Core operations + Codification) + Core operations (AD) +* $(1 - x) \cdot \left(\frac{nr \cdot nc}{BS} - n_f\right) \cdot$ *Gram–Schmidt (Non-anomalous) +* $x \cdot \left(\frac{nr \cdot nc}{BS} - n_f\right) \cdot$ *Gram–Schmidt (Anomalous)*. For clarity, some numerical results are given in Section 3.3.3.

3. Results

3.1. Reference Hyperspectral Data

For evaluating the performance and effectiveness of the proposed algorithm, we need a reference hyperspectral data set that permits evaluating both the lossy compression of HSIs and the detection of anomalies within them. Moreover, in order to evaluate our proposal in real scenarios, we have employed a new data set using real hyperspectral data collected by one of our unmanned aerial platforms. In any case, we have tested the goodness of the Gram–Schmidt method using conventional hyperspectral data sets in previous works for lossy compression and anomaly detection in [41,43], respectively. Despite the set of core operations proposed in this work has been optimized for allowing the concurrent execution of both processes, the obtained results are exactly the same that if both were independently executed using the already published algorithms.

The acquisition system used for sensing the data set carries a *Specim FX10* [53] push-broom hyperspectral camera mounted on a DJI Matrice 600 drone [54]. It covers the VNIR (Visible Near Infrared) range of the electromagnetic spectrum and collects information from 400 to 1000 nm using 224 spectral bands, with a spectral full width at half maximum (FWHM) of 5.5 nm, and 1024 spatial pixels per hyperspectral frame or scanned cross-track line. An extensive analysis of this acquisition system and its performance can be found in [9]. In this work, the first 20 and the last 45 spectral bands are discarded due to the low spectral response of the hyperspectral sensor at those wavelengths, what results in just 160 spectral bands being retained. Figure 2 graphically shows the sensor spectral response as well as the range of spectral bands selected for the experiments.
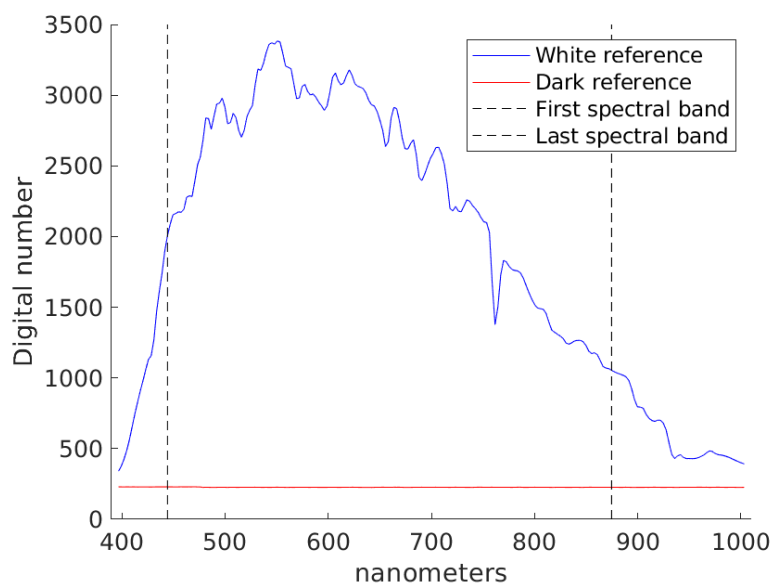

Figure 2. Spectral response of the Specim FX10 hyperspectral camera. The range of wavelengths collected by the data set employed in this manuscript is enclosed between dashed lines.

The data used for the experiments were collected by the aforementioned acquisition platform over different farming areas on the island of Gran Canaria, displayed in the Google Map picture shown in Figure 3a, during three different flight campaigns. The first one, highlighted in blue color, was carried out over a plantation of bananas in the south-west of the island, concretely in a village called Veneguera. The exact coordinates of the terrain are 27°52′17.4″N 15°45′44.2″W. The flight was performed at a height of 72 m over the ground at a speed of 6 m/s with the hyperspectral camera capturing frames at 125 frames per second (FPS), resulting in a ground sampling distance in line and across line of approximately 5 cm. This mission consisted of 6 waypoints that provided 3 swathes. The area covered during these swathes is highlighted in the Google Maps picture displayed in Figure 3b. Concretely,

three portions of 825 hyperspectral frames, with their 1024 hyperspectral pixels, where cut out from these swathes and used for the experiments. A RGB representation of these hyperspectral image locations within the corresponding swath are displayed in Figure 4a–c, Additionally, Figure 5a–c shows a closer view of the selected areas.


(a)

(b)

(c)

(d)
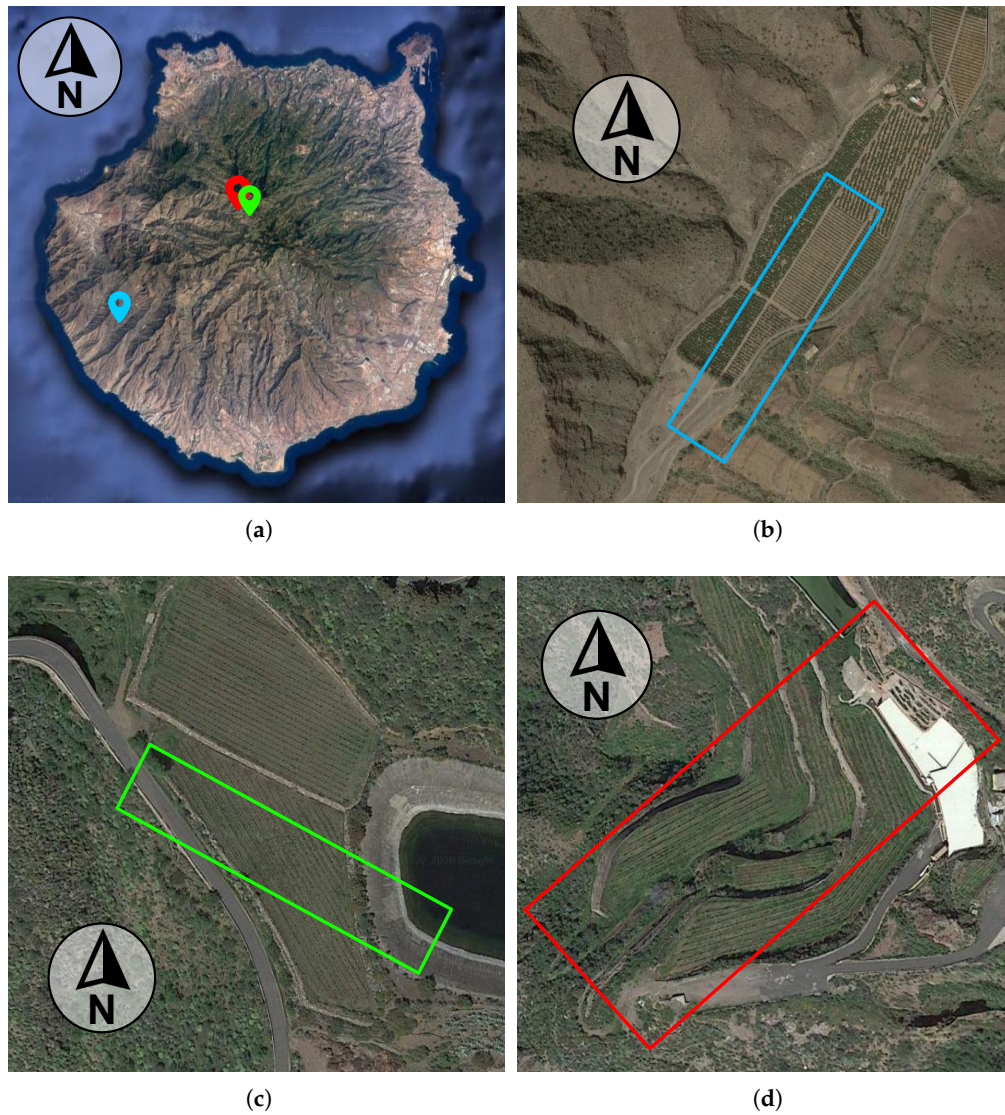
Figure 3. Google Maps pictures of the farming areas corresponding to the hyperspectral images that are used in this work. (**a**) Location of the terrains on the island of Gran Canaria. (**b**) Area covered during the first flight campaign over a banana plantation. (**c,d**) Area covered during the second and third flight campaigns over different vineyards.

(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)
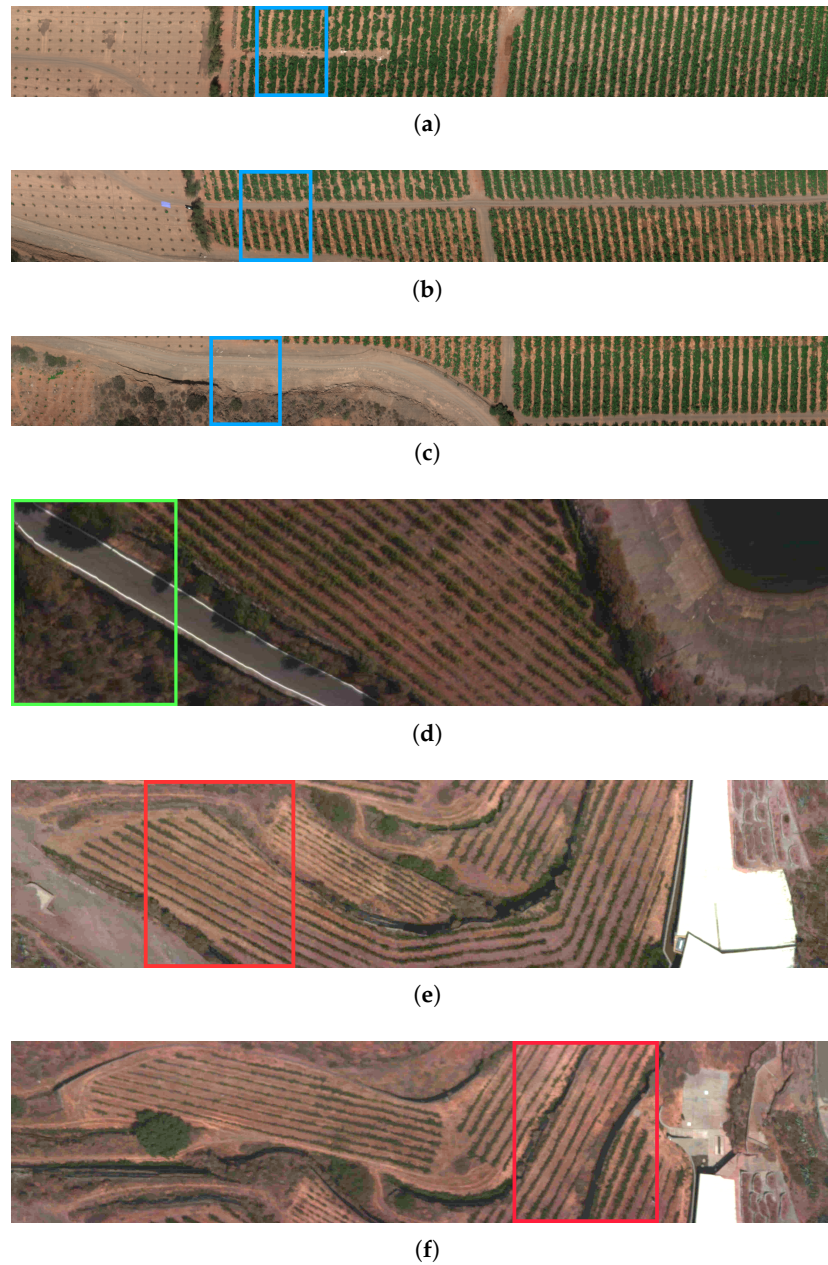
Figure 4. RGB representation of the hyperspectral data acquired in each mission campaign swath that was used in this work. Color squares highlight the regions selected for the experiments. (**a**) Image 1; (**b**) Image 2; (**c**) Image 3; (**d**) Image 4; (**e**) Image 5; (**f**) Image 6.
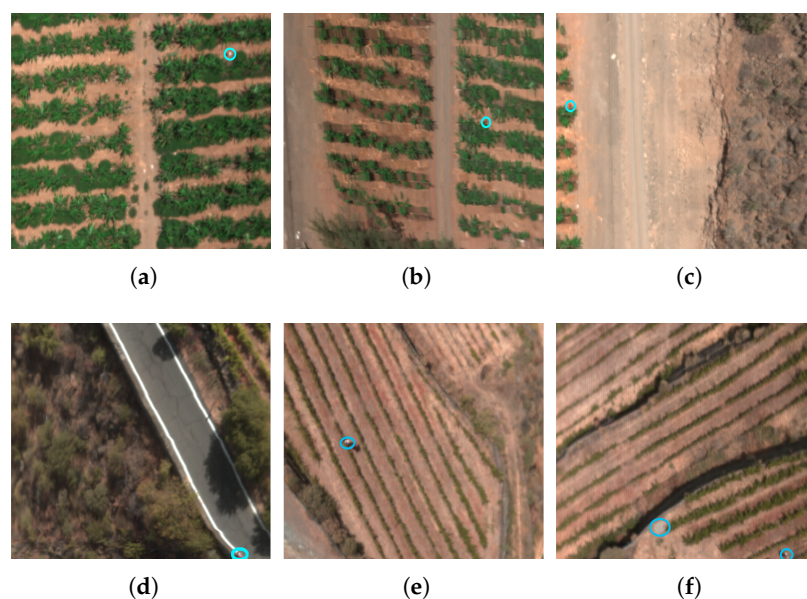
Figure 5. RGB representation of the employed test bench. Pixels enclosed in blue circles represent the anomalous entities to be detected. (**a**) Image 1; (**b**) Image 2; (**c**) Image 3; (**d**) Image 4; (**e**) Image 5; (**f**) Image 6.

The second flight campaign was carried out over a vineyard area in a village called Tejeda located in the center of the island. In particular, the exact coordinates of the scanned terrain are 27°59′35.6″N 15°36′25.6″W (highlighted in color green in Figure 3a). The flight was performed at a height of 45 m over the ground and at a speed of 4.5 m/s with the hyperspectral camera capturing frames at 150 FPS. This resulted in a ground sampling distance in line and across line of approximately 3 cm. This flight mission consisted of 12 waypoints that provided 6 swathes, but just one of them was used in the experiments carried out in this work. The ground area covered by this swath is highlighted in the Google Map picture displayed in Figure 3b. One smaller portion of 825 hyperspectral frames with all their 1024 hyperspectral pixels was cut out from the entire swath image for the simulations. Figure 5d displays its RGB representation while its location within the entire frame is shown in Figure 4d.

The third flight campaign was also carried out over a vineyard area in Tejeda. The exact coordinates of the scanned terrain, highlighted in color red in Figure 3a, are 27°59′15.2″N 15°35′51.9″W. This terrain was scanned in a flight performed at a height of 45 m over the ground, a speed of 6 m/s and the hyperspectral camera capturing at 200 FPS. The resulting ground sampling distance in line and across line was approximately 3 cm. The entire flight mission consisted of 5 swathes, but just 2 of them were used for the experiments in this work. The ground area covered by these swathes is highlighted in the Google Maps picture displayed in Figure 3d. From them, two smaller portions of 825 hyperspectral frames with all their 1024 hyperspectral pixels were cut out for the simulations. Figure 5e,f displays their RGB representations, while Figure 4e,f display their locations within the entire swathes.

The raw hyperspectral data collected by our system have been calibrated using a white and dark calibration to obtain reflectance values. Some examples of calibrated signatures corresponding to different pixels are displayed in Figure 6. We encourage the readers to read [9] where more details about this calibration can be found, as well as many other technical details related to the acquisition system. Nevertheless, we have not carried out either orthorectification or georeferencing process for the acquired data. The hyperspectral images used have been built up just by placing the subsequent captured hyperspectral frames one next to the other [55]. This does not degrade the quality of the experiments carried out in this work since the tested algorithms do not use any kind of spatial information. Finally, it is worth mentioning that all RGB representations obtained from the

hyperspectral images have been generated using the spectral bands corresponding to 450, 530 and 670 nm for the green, blue and red channels, respectively.

(a)	(b)
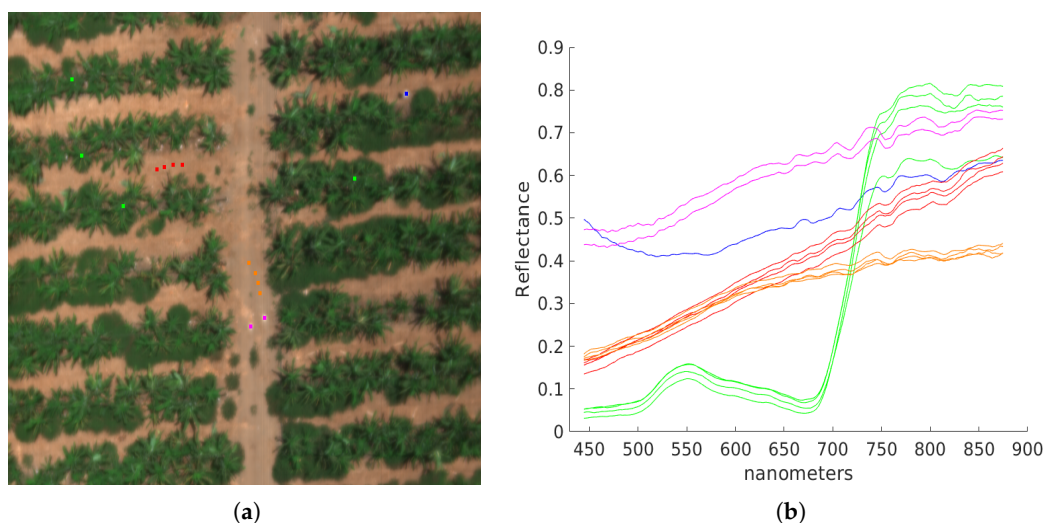
Figure 6. Example of spectral signatures corresponding to some real pixels of the first image used in the experiments (Figure 5a). (**a**) Pixel locations; (**b**) Pixel spectral signatures.

To evaluate the detection performance of the proposed method, the selected data set contains some artifacts that are considered to be anomalous pixels. Their locations within the test bench have been highlighted in Figure 5 using blue circles. These artifacts are people walking among the crop fields represented in Figure 5a–c,e. In the case of Figure 5d, the anomalous entities are two people standing next to the road. Finally, Figure 5f shows a person walking through the vineyards and a concrete construction as anomalous artifacts.

3.2. Evaluation Metrics

The goodness of the proposed method has been evaluated from two different perspectives, the quality of the compression performance and the efficiency in the detection of anomalous pixels.

Regarding the compression process, the test bench previously described has been compressed and decompressed using different configurations of the input parameters required for the compression process, which are *BS*, N_{bits} and *CR*. In order to evaluate the compression performance, the compression ratio achieved as well as the information lost in the compression-decompression process have been measured. The achieved compression ratio has been measured in two different ways, calculating the ratio between the original data volume and the compressed data volume and measuring the average number of bits per pixel per band, *bpppb*, used for representing the compressed image. Higher *CR* and lower *bpppb* values indicate that a higher compression has been achieved. The information lost in the compression-decompression process has been measured using five different quality metrics, which are: the *Signal-to-Noise Ratio (SNR)* (Equation (4)), the *Root Mean Squared Error (RMSE)* (Equation (5)), the *Peak Signal-to-Noise Ratio (PSNR)* (Equation (6)), the *Maximum Absolute Difference (MAD)* (Equation (7)) and the *Structural Similarity* (SSIM) index (Equation (8)). It is important to mention that although the proposed method independently compresses block of image pixels, $\mathbf{M}_K$, all these metrics have been calculated using the entire compressed–decompressed images. On this basis, *np* in Equations (4)–(8) represents the total number of pixels of the image under test and *Ic* the compressed–decompressed image.

The *MAD* evaluates the amount of information lost for the worst represented value of the entire image. Since the employed HSIs have been saved using 16 bits per pixel per band, the maximum possible (and worst) *MAD* value is 65,535. The *RMSE* evaluates the average information lost in

the entire image. Hence, higher *RMSE* values indicate higher amount of information lost in the compression-decompression process. The *SNR* and *PSNR* metrics also evaluate the average information lost in the entire image. However, these metrics compare the average error with the maximum possible error and hence, higher *SNR* and *PSNR* values indicate better compression performance. The *SNR* and *PSNR* metrics are calculated in decibels. *SSIM* measures distortions as a combination of three factors: loss of correlation, luminance distortion and contrast distortion. Hence, it is defined as the product of the powers of these three similarities, as it can be seen in Equation (8) where α, β and τ have been set to 1. Its dynamic range is $[-1,1]$ and value 1 indicates perfect structural similarity. Since *SSIM* is a quality assessment index originally designed for two-dimensional greyscale images, we have applied it in a band-by-band manner to evaluate the quality of the HSIs. Therefore, a mean *SSIM* index over all bands has been adopted in this manuscript [56].

$$\text{SNR} = 10 \cdot \log_{10}\left(\frac{\sum_{i=1}^{nb} \sum_{j=1}^{np} (I_{i,j})^2}{\sum_{i=1}^{nb} \sum_{j=1}^{np} (I_{i,j} - Ic_{i,j})^2} \right) \tag{4}$$

$$\text{RMSE} = \frac{1}{np \cdot nb} \cdot \sqrt{\sum_{i=1}^{nb} \sum_{j=1}^{np} (I_{i,j} - Ic_{i,j})^2} \tag{5}$$

$$\text{PSNR} = 10 \cdot \log_{10}\left(\frac{65535^2}{MSE} \right) \tag{6}$$

$$\text{MAD} = \max(I_{i,j} - Ic_{i,j}) \tag{7}$$

$$SSIM = \frac{1}{nb} \sum_{i=1}^{nb} \left[\left(\frac{2 \cdot \bar{I}_i \cdot \overline{Ic_i} + C1}{\bar{I}_i^2 + \overline{Ic_i}^2 + C1} \right)^\alpha \cdot \left(\frac{2 \cdot S_{I_i} \cdot S_{Ic_i} + C2}{S_{I_i}^2 \cdot S_{Ic_i}^2 + C2} \right)^\beta \cdot \left(\frac{S_{IIc_i}^2 + C3}{S_{I_i} \cdot S_{Ic_i} + C3} \right)^\tau \right] \tag{8}$$

$$\bar{I} = \frac{1}{np} \sum_{j=1}^{np} I_j \quad \overline{Ic} = \frac{1}{np} \sum_{j=1}^{np} Ic_j \quad S_I = \frac{1}{np-1} \cdot \sum_{j=1}^{np} (I_j - \bar{I})$$

$$S_{Ic} = \frac{1}{np-1} \cdot \sum_{j=1}^{np} (Ic_j - \overline{Ic}) \quad S_{IIc}^2 = \frac{1}{np-1} \sum_{j=1}^{np} (I_j - \bar{I}) \cdot (Ic_j - \overline{Ic})$$

Regarding the anomaly detection process, the evaluation of the proposed method in this field is visually performed through the description of the binary maps provided by the proposed method results, where detected anomalous pixels are segmented from the background. This is because the test images have been sensed at high altitudes and the exact position of the anomalies in the field was not measured. As a consequence, anomalous entities cover a very small number of image pixels. Additionally, the pixels at the object borders are mixed with the background. For this reason, it is very difficult to establish precise boundaries and hence, generate accurate pixel-level ground-truths.

3.3. Evaluation of the Proposed Method

This section discloses the results obtained in experiments carried out with the purpose of evaluating the goodness of the proposed algorithm for the compression of HSIs and the detection of anomalous pixels. To this end, data sets and quality metrics described in Sections 3.1 and 3.2, respectively, are used.

3.3.1. Compression Performance

In this Section, different experiments have been done to evaluate the behavior of the proposed method to compress HSIs. To this end, different configurations of the input parameters of the compression process have been employed. Concretely, the N_{bits} parameter is set to 12 bits, the *BS*

parameter to 1024 pixels and the *CR* parameter to 12, 16 and 20. According to [41], for image dynamic ranges of 16 bits, as those employed in this work, the more suitable value for N_{bits} is 12 bits. Regarding *BS*, the observation platform described in Section 3.1 captured the HSIs in a line-by-line fashion where each sensed frame is composed of 1024 pixels. Hence, *BS* has been set to 1024 pixels in the experiments. Table 2 collects the achieved *CR* and the employed number of *bpppb* for reconstructing the compressed images. Higher *CR* and lower *bpppb* indicate higher data compression. Likewise, Tables 3 and 4 collect the quantification of the information lost in the compression-decompression process through the *SNR*, *RMSE*, *PSNR*, *MAD* and *SSIM* evaluation metrics. It is to be recalled that although the proposed method independently compresses block of image pixels, $\mathbf{M}_K$, all these metrics have been calculated using the entire images after being compressed–decompressed.

Table 2. Compression Results. Achieved CR and bppp for the six data sets.

Inputs			Image 1		Image 2		Image 3		Image 4		Image 5		Image 6	
Nbits	BS	CR	CR	bpppb	CR	bpppb	CR	bpppb	CR	bpppb	CR	bpppb	CR	bpppb
		12	15.87	1.01	15.99	1.00	15.97	1.00	17.05	0.94	16.13	0.99	15.73	1.02
12	1024	16	22.56	0.71	22.81	0.70	22.82	0.70	24.43	0.65	22.83	0.70	22.16	0.72
		20	28.47	0.56	28.88	0.55	28.98	0.55	31.05	0.56	28.75	0.55	27.78	0.58

Table 3. Compression Results. Achieved SNR, MAD, SSIM, RMSE and PSNR for Image 1, Image 2 and Image 3.

Inputs			Image 1					Image 2					Image 3				
Nbits	BS	CR	SNR	MAD	SSIM	RMSE	PSNR	SNR	MAD	SSIM	RMSE	PSNR	SNR	MAD	SSIM	RMSE	PSNR
		12	45.87	1235.00	0.998	118.89	54.83	45.53	1318.00	0.998	110.46	55.46	45.72	946.00	0.998	96.43	56.65
12	1024	16	43.88	1504.00	0.998	149.46	52.84	43.76	1621.00	0.998	135.52	53.69	44.39	1170.00	0.998	112.42	55.31
		20	42.66	2245.00	0.998	171.96	51.62	42.67	1817.00	0.997	153.62	52.60	43.54	1343.00	0.997	123.93	54.47

Table 4. Compression Results. Achieved SNR, MAD, SSIM, RMSE and PSNR for Image 4, Image 5 and Image 6.

Inputs			Image 4					Image 5					Image 6				
Nbits	BS	CR	SNR	MAD	SSIM	RMSE	PSNR	SNR	MAD	SSIM	RMSE	PSNR	SNR	MAD	SSIM	RMSE	PSNR
		12	34.99	1340.00	0.997	121.11	54.67	35.43	2429.00	0.990	239.26	48.75	34.79	2936.00	0.988	281.37	47.34
12	1024	16	34.40	1557.00	0.997	129.54	54.08	34.84	2682.00	0.989	256.09	48.16	34.06	3496.00	0.987	305.97	46.62
		20	34.00	1694.00	0.997	135.75	53.67	34.43	2916.00	0.988	268.47	47.75	33.55	3799.00	0.985	324.68	46.10

According to results shown in Tables 2–4, it can be observed that the proposed method is able to reach very high compression ratios with a good rate-distortion for all tested configurations. As it can be seen in Table 2, we can conclude that the achieved *CR* values are always higher than the minimum desired *CR* established as input parameter. In particular, the achieved *CR* values are a bit higher for Image 4, Image 5 and Image 6 than for Image 1, Image 2 and Image 3.

It is also demonstrated that the proposed method can provide very high compression ratios with high rate-distortion ratios. The lowest achieved *CR* for a desired minimum *CR* = 12 is 15.73 (Table 2, Image 6). In this case, the obtained *SNR* is 34.79, the *PSNR* is 47.34, the *MAD* is 4.48 % of the maximum possible value and the *RMSE* is 281.37. For a desired *CR* = 16, the lowest achieved *CR* is 22.56 (Table 2, Image 1) while the obtained *SNR* is 43.88, the *PSNR* is 52.84, the proportion of *MAD* with respect to the maximum is 2.30 % and the *RMSE* is 149.46. Finally, for a desired *CR* = 20, the lowest achieved *CR* is 27.78 (Table 2, Image 6), the obtained *SNR* is 33.55, the *PSNR* is 46.10, the *MAD* is 5.79% of the maximum possible value and the *RMSE* is 324.68. In terms of distortions, including both luminance and contrast masking terms, the results are very promising since the maximum and minimum values reached for the SSIM metric were 0.998 and 0.985 respectively, which are very close to 1, the ideal SSIM value.

In conclusion, these compression ratios have been achieved losing little information, as shown by the values obtained for the different quality metrics. It is also worth mentioning that the compression performance of the proposed method is very solid and steady for the six different images.

3.3.2. Anomaly Detection Performance

Considering that the anomaly detection process employs the p_c vectors extracted in the compression stage, as it was explained in Section 2.2.3, simulations have been carried out using the same configuration settings employed to assess the compression process in Section 3.3.1 ($BS = 1024$, $N_{bits} = 12$ and desired $CR = [12, 16, 20]$). With respect to the input parameters related with the anomaly detection process, α is set to 1% and n_f to 100 hyperspectral frames, $\mathbf{F}_i$, for all test cases.

Figure 7 shows the map of anomalous pixels obtained by the proposed method superimposed on a panchromatic representation of the scenes to be tested to ease the result interpretations. It should be mentioned that the results for $CR = [12, 16, 20]$ are the same for all the 3 configuration settings. In displays shown in Figure 7, lines in blue color represent the n_f frames employed to estimate the background distribution, green lines represent the hyperspectral frames, $\mathbf{F}_i$, free of anomalous pixels while red lines represent those frames identified by our proposal to be corrupted by anomalous pixels. In addition, those anomalous pixels detected by our proposal have been also highlighted enclosing them inside red circles. Compared to the RGB representations displayed in Figure 5, anomalous objects detected in Image 1, Image 4, Image 5 and Image 6 match those anomalous entities highlighted in Figure 5. For Image 2 and Image 3, the anomalous bodies to be detected have been successfully identified but also other sparse brightness present in the scenes, as it can be seen in Figure 7b,c. In general terms, the anomaly detection results are very accurate although the spatial resolution covered by anomalous entities is very low and hence, they are composed of mixed spectral signatures.
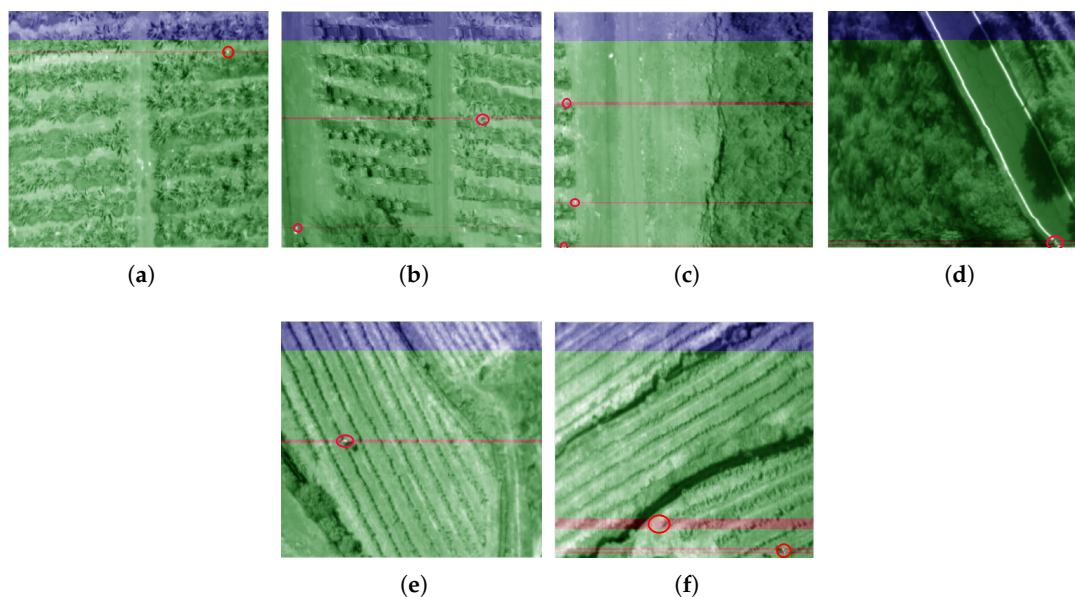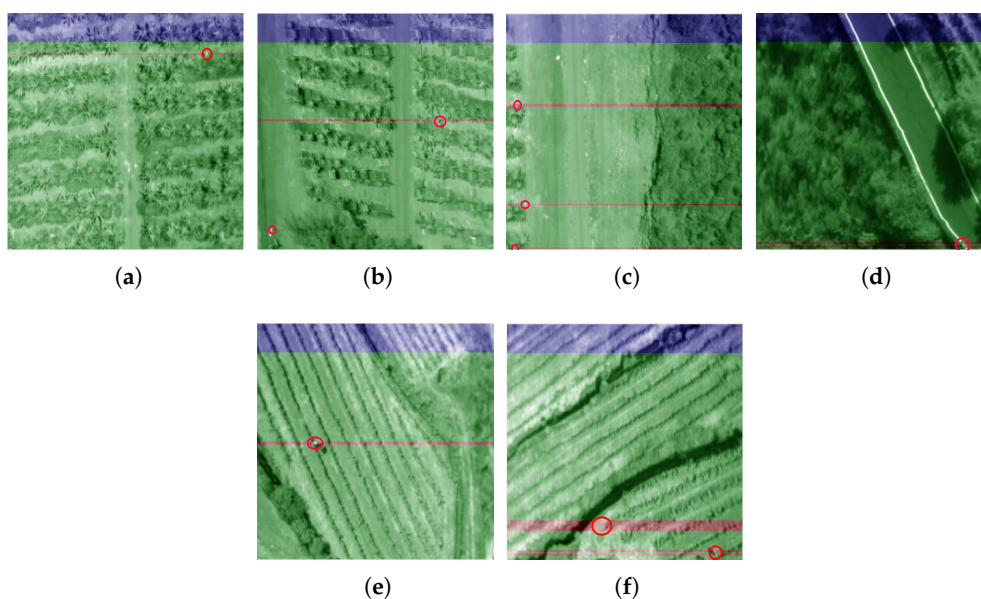

Figure 7. Anomaly detection results for input parameters $N_{bits} = 12$, $BS = 1024$ and $CR = [12,16,20]$. Lines in blue color represent the n_f frames employed to estimate the background distribution. Green lines represent the hyperspectral frames free of anomalies. Red lines represent those frames identified by our proposal to be corrupted by anomalous pixels. Pixels enclosed in red circles represent the anomalous pixels detected by our proposal. (**a**) Image 1; (**b**) Image 2; (**c**) Image 3; (**d**) Image 4; (**e**) Image 5; (**f**) Image 6.

As it can be concluded by results shown in Figure 7, one of the main advantages introduced by our proposal lies on the on-board execution of the anomaly detection process, which permits

the obtaining of a real-time performance and hence, a real-time identification of anomalous entities. Moreover, the fact that the processes to be executed are based on the same operations gives an extra advantage. Concretely, the results provided by the on-board execution of the anomaly detection process are exactly the same as those obtained by off-line performing it once that the compressed data packages are received and decompressed on the Earth surface. In order to clarify this comparison, we introduce Figure 8. Processes that follow the blue arrow refer to the proposed methodology where both compression and anomaly detection processes are performed on-board. On the contrary, processes on the red arrow refer to this last alternative where the hyperspectral frames are compressed on-board and the anomaly detection is performed on the Earth segment. Additionally, Figure 9 displays the map of anomalous pixels obtained by this last approach. As it can be seen, the obtained results are exactly the same as those provided by the proposed methodology, displayed in Figure 7, for any of the minimum desired CR values. Based on this assessment, we can conclude that the fact of using the same core operations in both compression and anomaly detection processes guarantee that the compression process does not affect the posterior anomaly detection performance when it is off-board executed using compressed–decompressed data. As a consequence, it permits tailoring to different scenarios that impose different requirements, ensuring the same results in all situations.
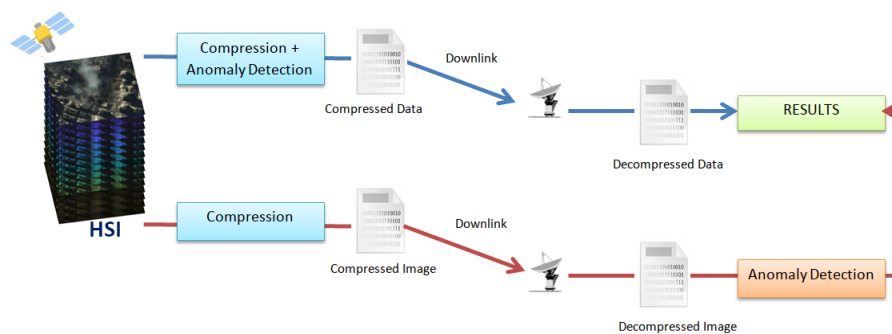
Figure 8. Block Diagram of the processes involved in the proposed method (blue arrow) versus those involved when images are on-board compressed, and the anomaly detection process is off-line performed (red arrow).


Figure 9. Anomaly detection results using compressed–decompressed images for input parameters $N_{bits} = 12$, $BS = 1024$ and $CR = [12, 16, 20]$. Lines in blue color represent the n_f frames employed to

estimate the background distribution. Green lines represent the hyperspectral frames free of anomalies. Red lines represent those frames identified by our proposal to be corrupted by anomalous pixels. Pixels enclosed in red circles represent the anomalous pixels detected by our proposal. (**a**) Image 1; (**b**) Image 2; (**c**) Image 3; (**d**) Image 4; (**e**) Image 5; (**f**) Image 6.

3.3.3. Computational Complexity

The greatest strength of our proposal is the ability to reuse operations when several hyperspectral analysis techniques are running in the same piece of hardware. Consequently, the number of operations to be executed considerably decreases compared to other approaches where not algorithmically related processes are implemented. For clarity, let us set some examples using an hypothetical data set with the same dimension as those images employed in this manuscript ($nr = 1024$, $nc = 1024$, $N_b = 160$). Since the presence of anomalous pixels is very unlikely, the probability of anomalous $\mathbf{M}_k$, x, is set to 0.010, p_{AD} has been rounded to 10 and p_c is estimated as a function of the minimum desired CR according to Equation (1).

In this context, Figure 10 displays the reduction of the number of FLOPs (in %) achieved by our proposal (blue arrow in Figure 8) when compared with the classical approach in which the lossy compression and the anomaly detection processes are executed as two independent algorithms (red arrow in Figure 8). This comparison has been done for different values of BS and minimum desired CR. As it can be seen, our proposal employs fewer FLOPs than if both processes were separately implemented. The reduction in the number of FLOPs changes with the configuration used, achieving a reduction of more than 50% in the best scenarios, and almost 35% in the worst one. Concretely, the largest computational burden reduction, in terms of FLOPs, is obtained when decreasing the block size used (BS) and increasing the minimum desired compression ratio (CR). The reduction in the number of operations to be executed is expected to be translated into a reduction in the required computational resources when following the methodology proposed in this work.
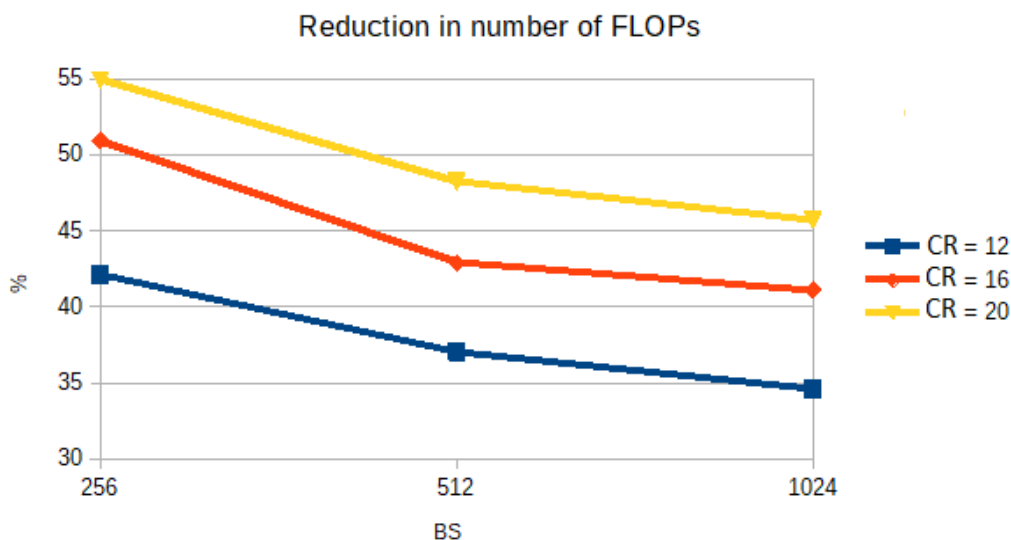
Figure 10. Reduction in the number of FLOPs (%) using the proposed methodology compared with the execution of both the lossy compression and the anomaly detection processes as two independent algorithms.

4. Discussions

In this work, we have approached the issue around the real-time processing of HSIs from a new algorithmic perspective. Concretely, we have proposed a set of common core operations that extracts

information from the HSIs useful for many hyperspectral analysis techniques. Therefore, this provides several benefits, above all in the field of on-board hyperspectral imaging processing when some time-sensitive applications must be executed in the same computing hardware device. First, it implies less time and effort during the stage of hardware acceleration since the same product can be reused for several algorithms targeting different applications. Secondly, it permits the execution of several tasks at the same time with the advantage of sharing the most computationally costly operations, thus reducing the overall computational cost and the required hardware resources. This last point has been demonstrated in this work from results shown in Figure 10. As it can be seen, using a common methodology for multiple analysis techniques could result in a reduction of more than 50% of hardware resources. This fact would cope with the resource limitations (power, size, weight and cost) of future space missions and emerging applications based on UAV.

In this context, we have particularly addressed the concurrent execution of lossy compression for HSIs and the detection of anomalous pixels. Presently, the new-generation sensors can produce continual or nearly continual stream of higher-dimensional data, which means that data volume to be managed has drastically increased. For instance, images used for experiments in this work were obtained in flight campaigns over large extensions of crop fields to monitor the plant status. The acquisition data rate of the hyperspectral camera carried by our aerial platform is up to 100 Mbytes per second, what results in almost 6 GB per minute. Accordingly, a 10 min flight could result in more than 50 GB. For this reason, the size of the acquired data must be drastically decreased for being able to rapidly transfer it, especially if real-time transmission is desired. On this basis, the high data rate provided by new-generation sensors imposes the necessity of carrying out a real-time compression process in order to efficiently store and/or transfer the acquired data without unnecessarily accumulating high amounts of uncompressed data. Additionally, high compression ratios are needed, but without losing essential information that could affect the posterior imagery analysis.

In the field of precision agriculture, there has been a true revolution arising from the application of new technologies to smart farming. Within the Agriculture 4.0 movement development, the European H2020 project ENABLE-S3 (European Initiative to Enable Validation for Highly Automated Safe and Secure Systems) introduced a case use about wildlife friendly agro-operation [47]. The goal of this work was the automated safe driving of harvesters in agriculture fields. To do this, image frames sensed by the acquisition system described in Section 3.1 were processed for detecting big obstacles (animals, rocks, among others) in the route of the harvester in order to prevent fatal accidents. In this context, anomaly detection is a good solution for detecting these obstacles since they represent entities that spectrally differ significantly from the background pattern. Additionally, based on this definition, anomaly detection analysis can be also used to detect some abnormal or suspicious behaviors in a lot of applications, such as precision agriculture (pest and diseases in the crops), environmental monitoring (flooding and hot spots), homeland security (gas leakages, illegal constructions), among others.

In this regard, if background spectral signatures are obtained in previous flights, changes in the land surface could be detected as anomalous entities by measuring the differences among abundances between bitemporal HSIs [57,58]. On the contrary, when background pixels are not known a priori, pixels extracted from each image block, $\mathbf{M}_k$, can be seen as local endmembers. In this scenario, the proposed set of core operations could be executed in background using as input image the pool of local endmembers being extracted from each image block, $\mathbf{M}_k$. At the end, when all $\mathbf{M}_k$ have been analyzed, we can obtain the global endmembers that can be sent to the Earth surface where the abundances can be off-line estimated using the compressed/decompressed images.

Finally, we would like to highlight the existing trade-off between the causality in line-by-line approaches and how to model the background distribution, which is the most important part in any adopted solution for anomaly detection. Actually, very few publications are made in this field where the anomaly detection issue is addressed in a line-by-line fashion [43,47,59,60]. In the solution proposed in this manuscript, the background distribution is estimated from several of the first sensed

hyperspectral image blocks, n_f, under the assumption that they are free of anomalous signatures and hence, fully representative of the background distribution. On this basis, enough n_f hyperspectral frames must be taken to ensure that all the spectral variability is covered and therefore generate a truthful background model. However, this methodology could be not a feasible solution in very heterogeneous scenarios, which may be one limitation of our proposal. In these cases, images could be compressed using the compression algorithm and then being off-line processed to detect the anomalous pixels.

To sum up, the methodology proposed in this work introduces the following advantages:

- High compression ratios and competitive rate-distortion compression performance where the rarest objects are preserved. Other state-of-the-art solutions for hyperspectral lossy compression behave as low-pass filters and tend to remove atypical elements [61–65].
- Line-by-Line performance. Since blocks of image pixels can be independently processed without any alignment required, it avoids accumulating high amount of uncompressed data. Hence, it permits reducing the amount of data to be processed and transferred. As a consequence, the proposed method becomes an ideal solution for applications under tight latency constraints or with limited available resources, such as memory, power and computational capabilities. In addition, the proposed algorithm fulfills the requirements imposed by applications based on push-broom/whisk-broom scanners, which permits commencement of the compression and anomaly detection processes as soon as a block of pixels is sensed.
- Low computational complexity and high level of parallelism of the most computationally demanding operations of the algorithm. This eases the hardware implementation of the proposal and reduces the amount of required hardware resources.
- Reduction in the employed hardware resources. Since the proposed methodology is based on a set of core operations common to several processes, the amount of hardware resources needed for their execution are considerably less than if different state-of-the-art algorithms were independently implemented.
- Accurate detection performance. The detection results obtained by the proposed method, which is executed online together with the compression process, are exactly the same as those provided when both the lossy compression and anomaly detection processes are executed as two independent algorithms.

5. Conclusions

Traditionally, data gathered by remote-sensing platforms are downloaded to the ground segment where they are off-line processed. It introduces several limitations related with the bandwidth and latency of the communication and the available aboard data storage that seriously harm the effectiveness of real-time applications. Accordingly, real-time on-board processing has experienced an increasing interest within the remote-sensing field to mitigate these limitations. However, the high data rate of the next-generation hyperspectral scanners and the limited computational resources, power and storage space available on-board make increasingly difficult to perform on-board real-time hyperspectral image processing when multiple applications are running at the same time onto the same piece of hardware. To tackle this challenging issue, in this work we have introduced a new methodology that enables the coexistence of multiple applications at the same time, reducing the employed hardware resources as well as the execution times.

In particular, in this work we have verified the adequacy of the proposed methodology for the concurrent execution of the lossy compression of HSIs jointly with the detection of anomalous signatures. In addition, we have also proved the benefits of employing this methodology in terms of number of operations. Concretely, we have verified that roughly 50% fewer operations are carried out using our proposal than if both processes were separately implemented. Therefore, it can be inferred than the required hardware resources may be also reduced in the same proportion. Additionally, in order to evaluate the goodness of the proposed algorithm for the lossy compression and anomaly

detection in HSIs, different experiments have been done using real data taken by an unmanned aerial vehicle flying over different types of crop fields. These scenarios are very challenging since anomalous entities cover an extremely reduced number of image pixels.

Future research lines are focused on extending this methodology to other fields such as target detection, classification, change detection and dimensionality reduction.

Author Contributions: Conceptualization, M.D. and R.G.; Formal analysis, M.D. and R.G.; Funding acquisition, S.L.; Investigation, M.D.; Methodology, M.D. and R.G.; Project administration, S.L.; Resources, P.H., J.F.L. and R.S.; Software, M.D.; Visualization, M.D., R.G., P.H., S.L., J.F.L. and R.S.; Writing—Original draft, M.D., R.G., P.H. and S.L. All authors have read and agreed to the published version of the manuscript.

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

References

1. Goetz, A.F. Three decades of hyperspectral remote sensing of the Earth: A personal view. *Remote Sens. Environ.* **2009**, *113*, S5–S16. [CrossRef]
2. Richards, J.A. *Remote Sensing Digital Image Analysis*; Springer: Berlin/Heidelberg, Germany, 2013.
3. Birk, A.; Wiggerich, B.; Bülow, H.; Pfingsthorn, M.; Schwertfeger, S. Safety, security, and rescue missions with an unmanned aerial vehicle (UAV). *J. Intell. Robot. Syst.* **2011**, *64*, 57–76. [CrossRef]
4. 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]
5. Govender, M.; Chetty, K.; Bulcock, H. A review of hyperspectral remote sensing and its application in vegetation and water resource studies. *Water Sa* **2007**, *33*. [CrossRef]
6. Ghamisi, P.; Yokoya, N.; Li, J.; Liao, W.; Liu, S.; Plaza, J.; Rasti, B.; Plaza, A. Advances in Hyperspectral Image and Signal Processing: A Comprehensive Overview of the State of the Art. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 37–78. [CrossRef]
7. Lu, G.; Fei, B. Medical hyperspectral imaging: A review. *J. Biomed. Opt.* **2014**, *19*, 010901. [CrossRef]
8. Khan, M.J.; Khan, H.S.; Yousaf, A.; Khurshid, K.; Abbas, A. Modern trends in hyperspectral image analysis: A review. *IEEE Access* **2018**, *6*, 14118–14129. [CrossRef]
9. Horstrand, P.; Guerra, R.; Rodríguez, A.; Díaz, M.; López, S.; López, J.F. A UAV platform based on a hyperspectral sensor for image capturing and on-board processing. *IEEE Access* **2019**, *7*, 66919–66938. [CrossRef]
10. Plaza, A.; Benediktsson, J.A.; Boardman, J.W.; Brazile, J.; Bruzzone, L.; Camps-Valls, G.; Chanussot, J.; Fauvel, M.; Gamba, P.; Gualtieri, A.; et al. Recent advances in techniques for hyperspectral image processing. *Remote Sens. Environ.* **2009**, *113*, S110–S122. [CrossRef]
11. Ortiz, A.; Rodríguez, A.; Guerra, R.; López, S.; Otero, A.; Sarmiento, R.; De la Torre, E. A runtime-scalable and hardware-accelerated approach to on-board linear unmixing of hyperspectral images. *Remote Sens.* **2018**, *10*, 1790. [CrossRef]
12. Villafranca, A.G.; Corbera, J.; Martín, F.; Marchán, J.F. Limitations of hyperspectral earth observation on small satellites. *J. Small Satell.* **2012**, *1*, 19–29.
13. Valentino, R.; Jung, W.S.; Ko, Y.B. A Design and Simulation of the Opportunistic Computation Offloading with Learning-Based Prediction for Unmanned Aerial Vehicle (UAV) Clustering Networks. *Sensors* **2018**, *18*, 3751. [CrossRef]
14. Lopez, S.; Vladimirova, T.; Gonzalez, C.; Resano, J.; Mozos, D.; Plaza, A. The promise of reconfigurable computing for hyperspectral imaging onboard systems: A review and trends. *Proc. IEEE* **2013**, *101*, 698–722. [CrossRef]
15. George, A.D.; Wilson, C.M. Onboard processing with hybrid and reconfigurable computing on small satellites. *Proc. IEEE* **2018**, *106*, 458–470. [CrossRef]

16. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]

17. Lu, K.; Xie, J.; Wan, Y.; Fu, S. Toward UAV-Based Airborne Computing. *IEEE Wirel. Commun.* **2019**, *26*, 172–179. [CrossRef]

18. Interuniversity Microelectronics Centre (IMEC). Hyperspectral Drone Camera System for Application Development. Available online: https://www.imec-int.com/drupal/sites/default/files/inline-files/UAV%20SNmosaic%20VIS%2BNIR%20hyperspectral%20imaging%20camera.pdf. (accessed on 13 April 2020).

19. Fu, S.; Chang, R.; Couture, S.; Menarini, M.; Escobar, M.; Kuteifan, M.; Lubarda, M.; Gabay, D.; Lomakin, V. Micromagnetics on high-performance workstation and mobile computational platforms. *J. Appl. Phys.* **2015**, *117*, 17E517. [CrossRef]

20. Díaz, M.; Guerra, R.; Horstrand, P.; Martel, E.; López, S.; López, J.F.; Roberto, S. Real-Time Hyperspectral Image Compression Onto Embedded GPUs. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 2792–2809. [CrossRef]

21. Plaza, A.; Du, Q.; Chang, Y.L.; King, R.L. High performance computing for hyperspectral remote sensing. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2011**, *4*, 528–544. [CrossRef]

22. Alcolea, A.; Paoletti, M.E.; Haut, J.M.; Resano, J.; Plaza, A. Inference in Supervised Spectral Classifiers for On-Board Hyperspectral Imaging: An Overview. *Remote Sens.* **2020**, *12*, 534. [CrossRef]

23. Du, Q.; Nekovei, R. Fast real-time onboard processing of hyperspectral imagery for detection and classification. *J. Real-Time Image Process.* **2009**, *4*, 273–286. [CrossRef]

24. Lentaris, G.; Maragos, K.; Stratakos, I.; Papadopoulos, L.; Papanikolaou, O.; Soudris, D.; Lourakis, M.; Zabulis, X.; Gonzalez-Arjona, D.; Furano, G. High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation. *J. Aerosp. Inf. Syst.* **2018**, *15*, 178–192. [CrossRef]

25. Orlandić, M.; Fjeldtvedt, J.; Johansen, T.A. A Parallel FPGA Implementation of the CCSDS-123 Compression Algorithm. *Remote Sens.* **2019**, *11*, 673. [CrossRef]

26. Plaza, A.J. Special issue on architectures and techniques for real-time processing of remotely sensed images. *J. Real-Time Image Process.* **2009**, *4*, 191–193. [CrossRef]

27. Ratle, F.; Camps-Valls, G.; Weston, J. Semisupervised neural networks for efficient hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2010**, *48*, 2271–2282. [CrossRef]

28. Christophe, E. Hyperspectral data compression tradeoff. In *Optical Remote Sensing*; Springer: Cham, Switzerland, 2011; pp. 9–29.

29. Hussain, A.J.; Al-Fayadh, A.; Radi, N. Image compression techniques: A survey in lossless and lossy algorithms. *Neurocomputing* **2018**, *300*, 44–69. [CrossRef]

30. Motta, G.; Rizzo, F.; Storer, J.A. *Hyperspectral Data Compression*; Springer Science & Business Media: Cham, Switzerland, 2006.

31. Horstrand, P.; López, J.F.; López, S.; Leppälampi, T.; Pusenius, M.; Rooker, M. A Simulation Environment for Validation and Verification of Real Time Hyperspectral Processing Algorithms on-Board a UAV. *Remote Sens.* **2019**, *11*, 1852. [CrossRef]

32. Su, H.; Du, P.; Du, Q. Semi-supervised dimensionality reduction using orthogonal projection divergence-based clustering for hyperspectral imagery. *Opt. Eng.* **2012**, *51*, 111715. [CrossRef]

33. Bioucas-Dias, J.M.; Plaza, A.; Dobigeon, N.; Parente, M.; Du, Q.; Gader, P.; Chanussot, J. Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2012**, *5*, 354–379. [CrossRef]

34. Chang, C.; Xiong, W.; Chen, H.; Chai, J. Maximum Orthogonal Subspace Projection Approach to Estimating the Number of Spectral Signal Sources in Hyperspectral Imagery. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 504–520. [CrossRef]

35. Bernabe, S.; Lopez, S.; Plaza, A.; Sarmiento, R.; Rodriguez, P.G. FPGA Design of an Automatic Target Generation Process for Hyperspectral Image Analysis. In Proceedings of the IEEE 17th International Conference on Parallel and Distributed Systems, Tainan, Taiwan, 7–9 December, 2011, pp. 1010–1015.

36. Li, H.; Chang, C. Linear spectral unmixing using least squares error, orthogonal projection and simplex volume for hyperspectral images. In Proceedings of the 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Tokyo, Japan, 2–5 June 2015, pp. 1–4. [CrossRef]

37. Kwon, H.; Nasrabadi, N.M. Kernel orthogonal subspace projection for hyperspectral signal classification. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 2952–2962. [CrossRef]

38. Ren, H.; Chang, C.-I.. Automatic spectral target recognition in hyperspectral imagery. *IEEE Trans. Aerosp. Electron. Syst.* **2003**, *39*, 1232–1249. [CrossRef]

39. Saad, Y. *Numerical Methods for Large Eigenvalue Problems*; Manchester University Press: Manchester, UK, 1992.

40. Guerra, R.; Santos, L.; López, S.; Sarmiento, R. A new fast algorithm for linearly unmixing hyperspectral images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 6752–6765. [CrossRef]

41. Guerra, R.; Barrios, Y.; Díaz, M.; Santos, L.; López, S.; Sarmiento, R. A New Algorithm for the On-Board Compression of Hyperspectral Images. *Remote Sens.* **2018**, *10*, 428. [CrossRef]

42. Díaz, M.; Guerra, R.; López, S.; Sarmiento, R. An algorithm for an accurate detection of anomalies in hyperspectral images with a low computational complexity. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1159–1176. [CrossRef]

43. Díaz, M.; Guerra, R.; Horstrand, P.; López, S.; Sarmiento, R. A Line-by-Line Fast Anomaly Detector for Hyperspectral Imagery. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 8968–8982. [CrossRef]

44. Guerra, R.; Barrios, Y.; Díaz, M.; Baez, A.; López, S.; Roberto, S. A Hardware-Friendly Hyperspectral Lossy Compressor for Next-Generation Space-Grade Field Programmable Gate Arrays. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 4813–4828. [CrossRef]

45. Sánchez, S.; Ramalho, R.; Sousa, L.; Plaza, A. Real-time implementation of remotely sensed hyperspectral image unmixing on GPUs. *J. Real-Time Image Process.* **2015**, *10*, 469–483. [CrossRef]

46. Gonzalez, C.; Sánchez, S.; Paz, A.; Resano, J.; Mozos, D.; Plaza, A. Use of FPGA or GPU-based architectures for remotely sensed hyperspectral image processing. *Integration* **2013**, *46*, 89–103. [CrossRef]

47. Horstrand, P.; Diaz, M.; Guerra, R.; Lopez, S.; Lopez, J.F. A Novel Hyperspectral Anomaly Detection Algorithm for Real-Time Applications With Push-Broom Sensors. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 4787–4797. [CrossRef]

48. Diaz, M.; Guerra Hernández, R.; Lopez, S. A Novel Hyperspectral Target Detection Algorithm For Real-Time Applications With Push-Broom Scanners. In Proceedings of the 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019, pp. 1–5. [CrossRef]

49. Nasrabadi, N.M. Hyperspectral target detection: An overview of current and future challenges. *IEEE Signal Process. Mag.* **2014**, *31*, 34–44. [CrossRef]

50. Consultative Committee for Space Data Systems (CCSDS). Blue Books: Recommended Standards. Available online: https://public.ccsds.org/Publications/BlueBooks.aspx (accessed on 23 March 2019).

51. Howard, P.G.; Vitter, J.S. Fast and efficient lossless image compression. In Proceedings of the Data Compression Conference, DCC'93, IEEE, Snowbird, UT, USA, 30 March–2 April 1993; pp. 351–360.

52. Díaz, M.; Guerra, R.; López, S. A Hardware-Friendly Anomaly Detector for Real-Time Applications With Push-Broom Scanners. In Proceedings of the 10th Workshop on Hyperspectral Imaging and Signal Processing: Evolution in Remote Sensing (WHISPERS), Amsterdam, The Netherlands, 24–26 September 2019; pp. 1–5. [CrossRef]

53. Specim FX1 Series Hyperspectral Cameras. Available online: http://www.specim.fi/fx/ (accessed on 23 March 2019).

54. DJI, MATRICE 600 PRO. Available online: https://www.dji.com/bg/matrice600 (accessed on 23 March 2019).

55. Guerra, R.; Horstrand, P.; Rodríguez, A.; Díaz, M.; Morales, A.; Jiménez, A.; López, S.; López, J.F. Optimal UAV movement control for farming area scanning using hyperspectral pushbroom sensors. In Proceedings of the XXXIV Conference on Design of Circuits and Integrated Systems (DCIS), IEEE, Bilbao, Spain, 20–22 November 2019; pp. 1–6.

56. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]

57. Jafarzadeh, H.; Hasanlou, M. An Unsupervised Binary and Multiple Change Detection Approach for Hyperspectral Imagery Based on Spectral Unmixing. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2019**, *12*, 4888–4906. [CrossRef]

58. Jafarzadeh, H.; Hasanlou, M. Assessing and comparing the performance of endmember extraction methods in multiple change detection using hyperspectral data. In Proceedings of the International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences, Karaj, Iran, 12–14 October 2019.

59. Chang, C.I.; Chiang, S.S. Anomaly detection and classification for hyperspectral imagery. *IEEE Trans. Geosci. Remote Sens.* **2002**, *40*, 1314–1325. [CrossRef]

60. Zhao, C.; Deng, W.; Yan, Y.; Yao, X. Progressive line processing of kernel RX anomaly detection algorithm for hyperspectral imagery. *Sensors* **2017**, *17*, 1815. [CrossRef]

61. Aiazzi, B.; Alparone, L.; Baronti, S. Quality issues for compression of hyperspectral imagery through spectrally adaptive DPCM. In *Satellite Data Compression*; Springer: Cham, Switzerland, 2012; pp. 115–147.

62. Lee, C.; Lee, S.; Lee, J. Effects of lossy compression on hyperspectral classification. In *Satellite Data Compression*; Springer: Cham, Switzerland, 2012; pp. 269–285.

63. Garcia-Vilchez, F.; Muñoz-Marí, J.; Zortea, M.; Blanes, I.; González-Ruiz, V.; Camps-Valls, G.; Plaza, A.; Serra-Sagristà, J. On the impact of lossy compression on hyperspectral image classification and unmixing. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 253–257. [CrossRef]

64. Du, Q.; Ly, N.; Fowler, J.E. An operational approach to PCA+ JPEG2000 compression of hyperspectral imagery. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2013**, *7*, 2237–2245. [CrossRef]

65. Chang, C.I. *Hyperspectral Data Processing: Algorithm Design and Analysis*; John Wiley & Sons: New York, NY, USA, 2013.