A Novel Hyperspectral Anomaly Detection Algorithm for Real-Time Applications With Push-Broom Sensors

Pablo Horstrand, María Díaz, Raúl Guerra, Sebastián López, *Senior Member, IEEE*, and José Fco. López

Abstract—**Most practical hyperspectral anomaly detection (AD) applications require real-time processing for detecting complex targets from their background. This is especially critical in defense and surveillance domains, but also in many other scenarios, in which a rapid response is mandatory to save human lives. Dealing with such a high dimensionality of data requires the conception of new algorithms to ease the demanding computing performance. Push-broom scanning represents the mainstream in hyperspectral imaging, introducing added complexity to the equation as there is no information of future pixels. In this paper, a novel technique named line-by-line anomaly detection (LbL-AD) algorithm, is presented as a way of performing real-time processing with a push-broom sensor. The sensor has been mounted on an unmanned aerial vehicle, and the acquired images, together with others from the scientific literature and synthetic ones, have been used to extensively validate the proposed algorithm in terms of accuracy, based on different metrics and processing time. Comparisons with state-of-the-art algorithms were accomplished in order to evaluate the goodness of the LbL-AD, giving as a result an outstanding performance.**

Index Terms—**Anomaly detection (AD), hyperspectral imagery, onboard processing, push-broom sensor, unmanned aerial vehicle (UAV).**

I. Introduction

THE use of unmanned aerial vehicles (UAVs) equipped with multi/hyperspectral sensors has gained momentum in the last few years as a smart strategy for collecting data for inspection, surveillance, and monitoring in the areas of defense, security, environmental protection, and civil domains, among others. The advantages of these aerial platforms over Earth observation satellites or manned airborne platforms is that they represent a lower-cost approach with a more flexible revisit time, while providing better spatial and sometimes even better spectral imagery resolutions, which permits a deeper and a more accurate data analysis [1].

There are several applications that benefit from these aerial platforms. For instance, in the agriculture domain, the utilization of UAVs for the acquisition of remotely sensed data has experienced an exponential growth [2], due to their capacity of periodically monitoring the state of the crops, their growth, ground water content, and other valuable information for the farmers, potentially increasing productivity and decreasing the costs. This application shares a common pattern with many others that also make use of UAV platforms with multi/hyperspectral sensors: Images are first captured, then data are downloaded after the flight is completed, and finally, data are processed offline to provide a result hours or even days later.

Nevertheless, there exist other applications in which a prompt response is mandatory. This is the case of the application motivating this work, which is a particular use case within the European H2020 project ENABLE-S3 (European Initiative to Enable Validation for Highly Automated Safe and Secure Systems). The goal for this particular scenario is the automated driving of a harvester in an agricultural field performing the cropping, watering, or pesticide spraying tasks. Initially, a drone equipped with a hyperspectral camera inspects the terrain in order to generate a set of maps with different vegetation indices, which gives information related to the status of the crop. This information, together with the specific coordinates, is then sent to the harvester, which initiates its autonomous driving to the region of interest and starts its labor. In relation with this driving activity, much has been reported related to fatal accidents due to agricultural mowing operations, which injure or kill animals. Additionally, any collision with a big obstacle (animal or rock) means a repairing cost, which, in some cases could be prohibitive. In order to avoid these situations and promote a wildlife friendly agro operation, a scheme similar to the one shown in Fig. 1 is shown. The drone flies at an altitude h and some meters ahead of the harvester ($security\ distance$), which depends on the harvester speed. If the hyperspectral camera detects an obstacle, a $stop\&go$ signal is sent to the harvester, and depending on the characteristic of the obstacle (dynamic or static), the harvester will stop till the dynamic obstacle disappears, or start an avoidance maneuver if it is static.

Manuscript received February 11, 2019; revised May 23, 2019; accepted May 27, 2019. Date of publication June 20, 2019; date of current version February 4, 2020. This work was supported in part by the European Commission through the ECSEL Joint Undertaking under ENABLE-S3 Project under Grant 692455, in part by the Spanish Government through the Project ENABLE-S3 under Grant PCIN-2015-225 and Project PLATINO under Grant TEC2017-86722-C4-1-R, and in part by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información (ACIISI) of the "Conserjería de Economía, Industria, Comercio y Conocimiento" of the "Gobierno de Canarias," which is partly financed by the European Social Fund (FSE) [POC 2014-2020, Eje 3 Tema Prioritario 74 (85%)]. *(Corresponding author: Pablo Horstrand.)*

P. Horstrand, M. Díaz, S. López, and J. F. López are with the Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, 35017 Las Palmas, Spain (e-mail: phorstrand@iuma.ulpgc.es; mdmartin@iuma.ulpgc.es; seblopez@iuma.ulpgc.es; lopez@iuma.ulpgc.es).

R. Guerra is with the Integrated System Design Division (DSI), Institute of Applied Microelectronics (IUMA), University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain (e-mail: rguerra@iuma.ulpgc.es).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTARS.2019.2919911

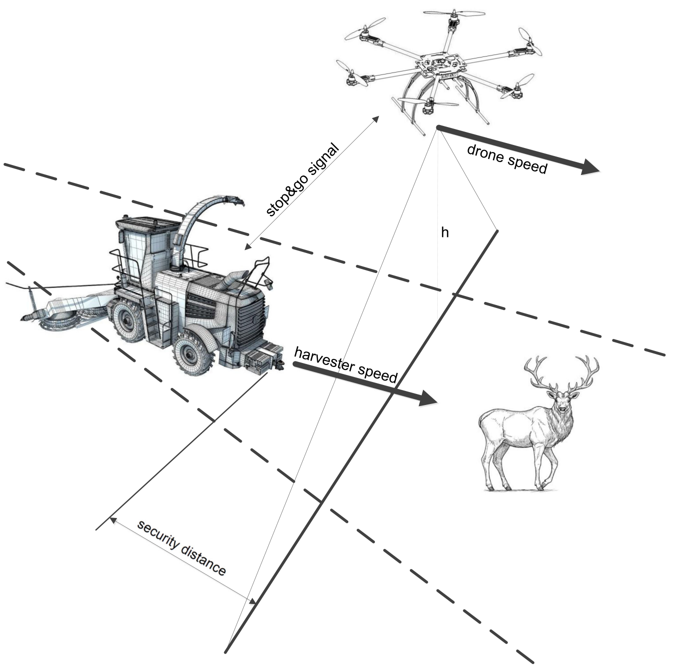


Fig. 1. Automated harvesting process, with the UAV a few meters in front of the harvester scanning the field ahead for obstacle detection.

The presented scenario entails some major challenges that are not overlooked by the authors and need to be further detailed. For instance, it is well known that the current autonomy of commercial UAV solutions powered by batteries is in the order of tens of minutes making the whole operation too discontinuous and therefore unattractive for potential users. There are nonetheless works in the scientific literature proposing different solutions for such problem. On one hand, a wireless power charge approach in order to reduce the stopping time [3]. On the other hand the use of alternative propulsion systems (or hybrid combinations of some of them), which alleviate the restrictions imposed by electric batteries in terms of autonomy, are attracting great attention from the scientific community [4]–[6]. Another important aspect to take into account when it comes to UAVs, being an autonomous vehicle, is the safety critical operation it implies. Regulations are rather strict and sometimes even present an additional economical burden. However, as we see in other domains, such as fire monitoring [7], construction hazards [8], or even photogrammetry under extreme weather conditions [9], the safety operation can be kept and at the same time profit the advantages of the use of such system.

There are different strategies that have been evaluated to sort the problems raised by the ENABLE-S3 Farming Use Case presented before, mainly based on thermal sensors [10] and LIDARs. When implementing such autonomous vehicle technologies, being at stake animals and human being lives, safety and security becomes a critical issue and therefore, the right approach is the combination of different technologies. Thermal approach may fail in warm environments where living beings are not so well distinguished anymore, and of course as well detecting inert objects. On the other hand, LIDARs might be alternatively installed in the harvesting vehicle or in the supervisory drone. When mounted on the harvester, an uneven terrain profile or the height of the crop can produce false positives, while mounting it on the drone has the drawback of additional weight and power consumption, reducing the whole system autonomy. In this sense, great effort has been carried out by the scientific community to reduce the sensor weight [11], [12]. For the mentioned reasons, a new approach is proposed in this work, consisting in using hyperspectral images (HSIs) that could complement the results obtained by the mentioned technologies. Since we are dealing with a homogeneous field with small objects coming up in the scene from time to time, anomaly detection (AD) has been selected as an alternative approach to tackle the problem. Such technique applied to hyperspectral data allows distinguishing rare objects with unknown spectral signatures that are not particularly abundant in a scene. This possibility of distinguishing a scarce group of pixels whose spectral signature significantly differs from their surroundings represents a crucial feature and has inspired the appearance of a huge amount of hyperspectral AD algorithms in the recent scientific literature.

The Reed–Xiaoli (RX) algorithm [13] is one of the first developments in this field, being viewed as a benchmark to which other methods are compared. The RX anomaly detector is based on the Mahalanobis distance between the pixel under test and the background class. It assumes that the background follows a single Gaussian normal distribution and the probability density function is used to classify pixels as part of the background class. Thus, the background mean and the inverse covariance matrix must be well estimated; otherwise, they could be contaminated by anomalies causing a subsequent misclassification. Several variations of the RX detection technique have been proposed in the literature in order to improve its performance. Subspace RX (SSRX) [14] and RX after orthogonal subspace projection (OSPRX) [15] are global anomaly detectors that apply principal component analysis (PCA) or singular value decomposition (SVD) to the datacube. The goal is to reduce the data volume to a smaller subspace where the first PCA/SVD bands are supposed to represent the background class. SSRX discards these bands, and then, RX is applied to the remaining subspace. On the contrary, OSPRX projects the data onto the orthogonal subspace before applying RX.

In the recent years, new approaches to the problem have emerged, in order to cope, for instance, with inadequate Gaussian-distributed representations for nonhomogeneous backgrounds [16], the presence of noise in the images, by using a combined similarity criterion anomaly detector method [17], as well as the removal of outliers by using a collaborative representation detector [18], [19]. Other authors have explored techniques extensively used in other machine vision domains as it is the use of dictionaries, to separate the anomalies from the background [20], and even combined the AD process with another very common processing stage in HSI, such as compression [21].

Unfortunately, all these algorithms require the sensing of the whole HSI before starting with the process of finding the anomalies in the captured scene. However, the most widely used sensors in remote sensing applications nowadays are based on push-broom hyperspectral scanners, in which the image is captured in a line-by-line fashion, since they provide an outstanding spectral resolution and take advantage of the movement of the UAV, aircraft, or satellite that carries them for capturing the whole hypercube. Hence, for applications under strict real-time constraints in which the captured images must be processed in a short period of time, it is more efficient if the anomalies are uncovered as soon as the hyperspectral data are sensed. Moreover, this kind of on-the-fly AD drastically reduces the vast amount of memory that is required on-board the sensing platform in order to store the entire hypercubes.

In this scenario, this paper proposes a novel hyperspectral AD algorithm specially conceived for being able to process the lines of pixels captured by a push-broom scanner as soon as they are sensed. More concretely, the proposed line-by-line anomaly detection (LbL-AD) algorithm is based, similar to the OSPRX algorithm, on the concept of orthogonal subspace projections, but employing a low computational processing chain that guarantees a precise detection of the anomalies present in a hyperspectral scene.

For the purpose of testing the performance of the algorithm presented in this paper, synthetic images as well as real images captured both by commercial aerial platforms and by our own UAV platform based on a Specim FX10 visible and near infrared range (VNIR) hyperspectral push-broom sensor [22] mounted on a DJI Matrice 600 drone [23], have been used.

The rest of the paper is organized as follows. Section II describes, step by step, the proposed LbL-AD algorithm. Section III presents the hyperspectral data and the assessment metrics utilized for comparing the performance given by the LbL-AD algorithm versus other state-of-the-art proposals. Section IV outlines the main results obtained in terms of detection performance and execution performance. Finally, Section V draws the most representative conclusions achieved in this work as well as further research directions.

II. LbL-AD Algorithm

As it has been already mentioned, the LbL-AD algorithm is an anomaly detection algorithm suitable for real-time applications using push-broom hyperspectral sensors whose main advantage is its fast computation combined with a good detection performance, being able to process each line of pixels as soon as they are sensed. This implies a reduction in the required memory resources since it is not necessary to store the full hypercube prior to its processing.

In order to keep a low computational complexity of the AD process, the proposed LbL-AD algorithm follows a twofold strategy: It starts processing the first n lines captured as a whole bunch and then it follows a progressive line-by-line processing for the rest of the lines of the image in which only a reduced amount of operations are performed in order to update the results for the new acquired line of pixels.

In particular, the following procedure is carried out for the first n lines captured by the sensor.

1) A reference average pixel is computed considering all the pixels contained in these lines.
2) The previously obtained average value is subtracted to each sensed pixel.
3) The covariance matrix corresponding to these pixels is calculated. At this point, it is worth to highlight that this covariance matrix is not divided by the total number of pixels, since in this way this matrix can be reutilized in the next iterations (next lines of pixels), adding to it the new covariance matrix of the acquired line. For this to work, the number of pixels so far processed needs to be accounted as well.
4) PCA is performed then onto the covariance matrix to find the d highest eigenvalues and their associated eigenvectors. Here, in order to keep the LbL-AD algorithm within a low computational burden, the following computing strategy is applied.
    a) First, the highest eigenvalue and its associated eigenvector are obtained by means of the power iteration method [24].
    b) Afterward, deflation [25] is performed onto the covariance matrix to calculate the next $d-1$ eigenvalues and eigenvectors, by means of successively applying the power iteration method. This process is repeated until the d desired number of components is obtained.

    The reason for having selected this combination of methods is dual: On one hand, it is a very fast method for just obtaining a few principal components, and on the other hand, if the algorithm is wisely initialized, the computation time can be significantly reduced. This last characteristic is exploited by the proposed LbL-AD algorithm, as for each iteration (each new line of pixels that is acquired after the first n lines), the subspace calculated from the previously sensed pixels is used for initializing the algorithm, which brings a significant speedup factor to the process.
5) The pixels are projected onto the subspace spanned by the d eigenvectors obtained in the previous step.
6) The Mahalanobis distance is calculated for each pixel, as it is done in the original RX algorithm. This step involves the calculation of the pseudoinverse of the covariance matrix. However, as far as the covariance matrix in the new projected subspace is a diagonal matrix, its inverse is obtained by just inverting its single elements individually, which is again a huge save in terms of computing time.
7) Based on the calculated distance result, it is decided whether each pixel is an anomaly or not. Thanks to the good separability between background and anomalies of the result provided by the algorithm, this task is carried out applying a simple technique of outlier detection.
8) The background pixels Mahalanobis distance mean value and standard deviation are computed for later use in the process.

Once these n lines of pixels have been processed, the following line-by-line procedure is carried out for the rest of the pixels in the hyperspectral scene under analysis.

A) The average pixel obtained previously in step 1 is subtracted to each pixel of the line under processing.
B) The covariance matrix of the line under processing is calculated and added to the existing covariance matrix calculated previously in step 3. As with step A, this is performed under the assumption that the average pixel obtained in

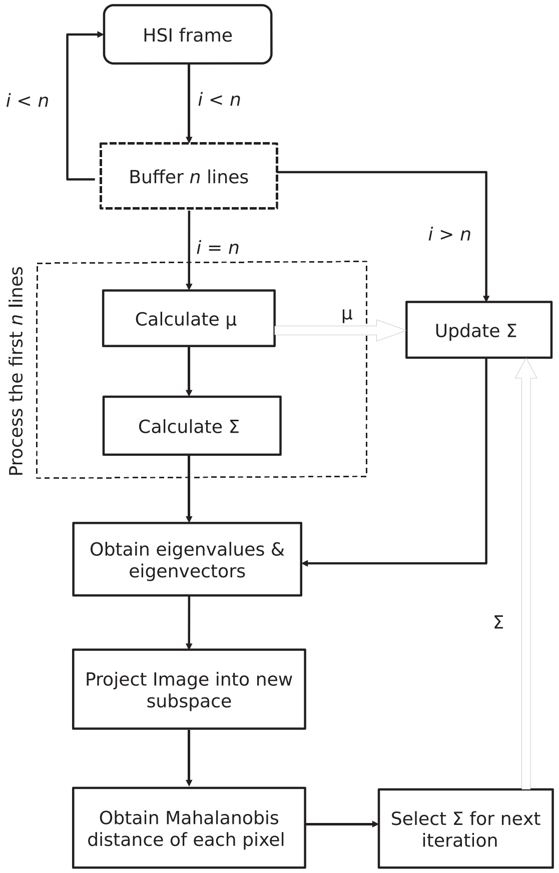

HSI frame
i < n
i < n
Buffer n lines
i = n
i > n
Process the first n lines
Calculate μ
μ
Update Σ
Calculate Σ
Obtain eigenvalues & eigenvectors
Σ
Project Image into new subspace
Obtain Mahalanobis distance of each pixel
Select Σ for next iteration

Fig. 2. General overview of the LbL-AD algorithm.

step 1 remains approximately constant, which allows to skip the full computation of a new covariance matrix for each new line of pixels. Moreover, this method allows us to keep in memory just the previous covariance matrix, but not the entire amount of pixels processed, which means a huge save in memory space and number of memory transactions.

C) Steps 4–7 are applied to the line of pixels under processing.

D) If in step 7, anomalies were detected, then the recently calculated covariance matrix is discarded, and the previous covariance matrix available recovered for processing the next line. The reason for doing this is to keep as much as possible the covariance matrix modeling only the background, so whenever an anomaly is coming into the scene it will be something new. This confides the result an homogeneity that otherwise it would not have.

E) Finally, previous step 8 is applied, exclusively to the background pixels.

Fig. 2 shows a general vision of the stages involved in the described algorithm. The following subsections will detail the different parts of the proposed algorithm.

A. Computation of First n Lines

The first n lines are buffered into a matrix M, which afterward is used to calculate the average pixel μ, to be subtracted to every processed image line. The covariance matrix Σ results from the obtained normalized matrix M', for which the eigenvectors $\vec{u_1}, \ldots, \vec{u_d}$ and the eigenvalues $s_1, \ldots, s_d$ are calculated. After that, the new subspace is calculated as explained in Section II-B. Algorithm 1 shows the pseudocode of the algorithm initialization, where r stands for each captured line of n_c pixels.

Algorithm 1: Computation of First n Lines.

```
Inputs: $\mathbf{M} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_n]$ where $\mathbf{r}_i = [\mathbf{s}_{i1}, \mathbf{s}_{i2}, ..., \mathbf{s}_{inc}]$;
        {Initial portion of the HSI}
$\mathbf{d}$, {Number of principal components to be selected}
$\mathbf{N} = \mathbf{n} \times \mathbf{n}_c$, {Number pixels contained in the first n lines}
$\mu = mean(\mathbf{M})$; {Average pixel}
for $i = 1$ to $n$ do
    for $j = 1$ to $nc$ do
        $\mathbf{s}_{ij} = \mathbf{s}_{ij} - \mu$;
    end
end
$\mathbf{\Sigma} = \mathbf{M}^T\mathbf{M}$;
$\mathbf{M'} = \frac{\mathbf{\Sigma}}{\mathbf{N}}$;
$[Psi, S] = PCA(\mathbf{M'}, \mathbf{d})$;
Outputs:
$\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{S}$, {Eigenvalues}
PSI, {Eigenvectors}
$\mathbf{n}$, {Number of processed pixels}
$\mu$, {Average pixel}
```

B. Subspace Calculation. Power Iteration Method and Deflation.

On the first run, the subspace is calculated for the computed covariance matrix Σ, from the buffered n lines and then it is updated on each iteration after the covariance matrix has been updated. Since in the first iteration there are no previous eigenvectors for feeding the algorithm, the input eigenvectors PSI_{in} are initialized with random values. However, on the subsequent iterations the set of eigenvectors defining the previous subspace are introduced as inputs of the algorithm and used to initialize the calculation, considering that there shall not be a big difference between the set of eigenvectors computed from two consecutive iterations. The other two inputs to the algorithm are the number of eigenvalues and eigenvectors to be calculated d, and a stop condition for the convergence of the power iteration method, *tol*. An auxiliary matrix B is used to deflate the original covariance matrix and keep calculating subsequent eigenvalues and eigenvectors. The deflation process mainly consists in obtaining a new matrix that provides the same result than the original one in the orthogonal direction to it and it is zero otherwise. A detailed description of the subspace calculation process is presented in Algorithm 2.

The subspace algorithm makes a recurrent call to the $Power\ Method$ function, whose behavior is described in Algorithm 3, until all d eigenvalues and eigenvectors have been

Algorithm 2: Subspace Calculaton.

```
Inputs: $\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{PSI}_{in}$, {Input Eigenvectors }
d, {number of eigenvalues to be calculated}
tol, {stop condition for the power iteration method}
$[PSI_1, S_1] = power\ method(\Sigma, tol, PSI\_in_1)$;
$\mathbf{i = 2}$;
$\mathbf{B = \Sigma}$; {Support matrix to compute deflation}
while $i <= d$ do
    $\mathbf{B = B - PSI}_{i-1}^T\mathbf{PSI}_{i-1}$;
    $[PSI_i, S_i] = powermethod(\Sigma, tol, PSI\_in_i)$;
    $\mathbf{i = i + 1}$;
end
Outputs:
S, {Eigenvalues}
PSI, {Eigenvectors }
```

Algorithm 3: Power Method.

```
Inputs: $\mathbf{\Sigma}$, {Covariance matrix}
$\mathbf{b}_k$, {Input Eigenvector}
tol, {stop condition}
$\mathbf{rayl_0 = 0}$;
$\mathbf{rayl} = \frac{\mathbf{b}_k^T\mathbf{\Sigma b}_k}{\mathbf{b}_k^T\mathbf{b}_k}$;
while $abs(rayl_0 - rayl) > tol$ do
    $\mathbf{rayl_0 = rayl}$;
    $\mathbf{b}_{k+1} = \frac{\mathbf{\Sigma b}_k}{\|\mathbf{\Sigma b}_k\|}$;
    $\mathbf{rayl} = \frac{\mathbf{b}_k^T\mathbf{\Sigma b}_k}{\mathbf{b}_k^T\mathbf{b}_k}$;
end
Outputs:
rayl, {Calculated Eigenvalue}
$\mathbf{b}_k$, {Calculated Eigenvector}
```

Algorithm 4: Image Projection and RX.

```
Inputs: $\mathbf{r}_k = [\mathbf{s}_{k1}, \mathbf{s}_{k2}, ..., \mathbf{s}_{knc}]$ where $k > n$; {captured
    frame of nc pixels}
$\mathbf{\Sigma}\_in$, {Covariance matrix of the already processed
image}
$\mu$, {Average pixel}
$\mu_{dist\_in}, \sigma_{dist\_in}$, {Background distance mean and
standard deviation}
$\mathbf{B}_{mask\_in} = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_{k-1}]$, {Binary result of
processed pixels so far}
$\mathbf{L = k \times n}_c$, {Number of pixels processed so far, line $k$
    included }
for $j = 1$ to $nc$ do
    $\mathbf{s}_{kj} = \mathbf{s}_{kj} - \mu$;
end
$\mathbf{\Sigma = M}^T\mathbf{M} + \mathbf{\Sigma}\_in$;
$[PSI, S] = Algorithm\_2(\frac{\mathbf{\Sigma}}{\mathbf{L}}, \mathbf{PSI}, \mathbf{d}, \mathbf{tol})$;
$\mathbf{Y} = PSI^T \times M$; {Projected Image to the new calculated
    subspace}
$\mathbf{P = RX(s) = s}^T\mathbf{S}^{-1}\mathbf{s}$
update $= 1$
for $j = 1$ to $nc$ do
    if $\boldsymbol{p}_j > \ \mu_{dist\_in} + \sigma_{dist\_in} * 15$ then
        $\mathbf{b}_{n+j} = 1$;
        update $= 0$;
    else
        $\mathbf{b}_{n+j} = 0$;
    end
end
$\mu_{dist} = \frac{\sum_{i=1}^{nc} p_i * b_i}{L}$
$\sigma_{dist} = \frac{\sqrt{\sum_{i=1}^{nc}(p_i - \mu_{dist})^2 * b_i}}{L-1}$;
Outputs:
P {Vector of Mahalanobis distances}
$\mathbf{B}_{mask}$ {Binary result of processed pixels}
PSI, {Eigenvectors}
$\mu_{dist}, \sigma_{dist}$, {Bck distance mean and std deviation}
if $update$ then
    $\mathbf{\Sigma}$, {Covariance matrix};
else
    $\mathbf{\Sigma}\_in$, {Input covariance matrix};
end
```

obtained. As explained earlier, the initialization with the previous obtained eigenvector is used to speed up convergence, which combined with the use of the Rayleigh coefficient *rayl* has been proven to be very efficient [26]. This coefficient is calculated using the eigenvector approximation and represents in its turn a very good approximation of the eigenvalue. The method consists in iteratively calculating the new coefficient *rayl* with the recently obtained vector and comparing it with the value calculated in the previous iteration $rayl_0$ until the difference between them is lower than the input tolerance *tol*.

C. Image Projection and Distance Calculation

After the first n lines have been processed, the proposed algorithm processes the sensed data line by line, as it is described in Algorithm 4. As it has been already mentioned, the average pixel μ obtained in the first stage of the algorithm, is subtracted from each pixel, and then, the covariance matrix is updated to obtain Σ_{new} and the new set of d eigenvalues and eigenvectors from the updated Σ_{new}. The process then continues projecting the image onto the new subspace and obtaining the Mahalanobis distance for each pixel. Finally, the mean value μ_{dist} and the standard deviation σ_{dist} of the Mahalanobis distance of just the background pixels are calculated and updated on every iteration, in order to account for possible outliers, and thus, anomalies. In case anomalies are found, the covariance matrix of the previous iteration is used, instead of the updated one as long as it is not the first iteration. As it can be seen in the pseudocode, a very large value, for instance, 15 times the standard deviation is used to be very conservative and avoid covariance update in case anomalies are found. The reason for updating the covariance matrix selectively is to obtain a more homogeneous result in terms of the anomalies.

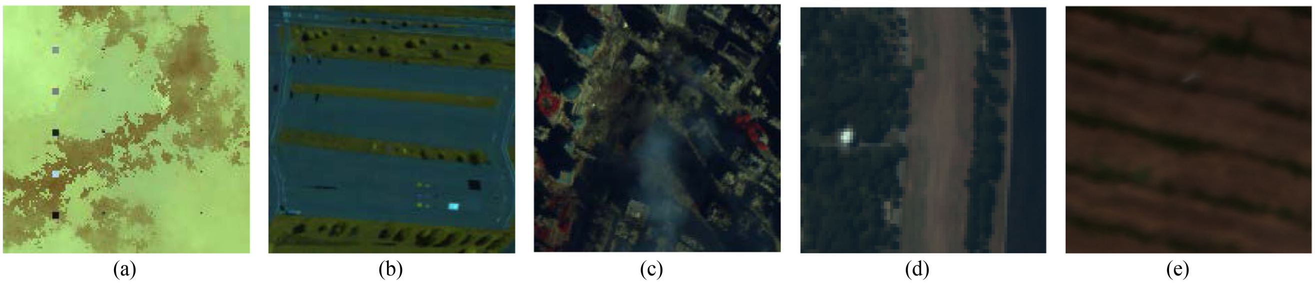


Fig. 3. Test datasets. (a) Synthetic image. (b) WASP RIT scene. (c) AVIRIS WTC scene. (d) HYDICE WDC scene. (e) UAV scene.

III. Data and Evaluation Metrics

A. Reference Hyperspectral Data

In this paper, both simulated and real HSIs have been used to evaluate the performance and effectiveness of the proposed method. The simulated data have a size of 150 × 150 pixels and 429 spectral bands. It was generated using a spectral library collected from the United States Geological Survey (USGS) [27]. Background was simulated using four different spectral signatures whose abundances were generated using a Gaussian spherical distribution [28]. Twenty panels of different sizes arranged in a 5 × 4 matrix were introduced as anomalies. There are five 4 × 4 pure-pixel panels lined up in five rows in the first column, five 2 × 2 mixed-pixel panels in the second column, five subpixel panels combined with the background in a proportion of 50% in the third column and five subpixel panels blended with the background at 75%. Therefore, the simulated image has 110 anomaly pixels, a 0.49% of the image, as it is shown in Fig. 3(a). This dataset is very challenging for AD because of the high spectral similarities between some anomalous and background signatures.

In order to test the proposed algorithm in a more realistic scenario, four real hyperspectral datasets have been also used: Three well-known datasets available in the literature, and a dataset captured by an UAV available at our lab. The first real dataset was taken over the Rochester Institute of Technology (RIT) by the Wildfire Airborne Sensor Program (WASP) Imaging System [29]. The system covers the visible, short, mid and long wave infrared regions of the spectrum. The sensor comprised a high-resolution color camera covering the visible spectrum, a short infrared imager covering 0.9–1.5 μm, mid infrared imager covering 3.0–5.0 μm, and a long infrared imager covering 8.0–9.2 μm. A portion of the overall image taken over a parking lot with a size of 180 × 180 pixels and 120 bands has been used in this study, as can be seen in Fig. 3(b) where anomalies are fabric targets which consist of 72 pixels and account for 0.22% of the image. The second real dataset was collected by the NASA Jet Propulsion Laboratory's Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS) over the World Trade Centre (WTC) area in New York City on September 16, 2001 [30]. The original dataset has a size of 614 × 512 pixels and 224 spectral bands from 0.4 to 2.5 μm although a smaller region with size of 200 × 200 pixels was selected as dataset. Anomalies are thermal hot spots, which consist of 83 pixels and account for 0.21% of the image scene. Fig. 3(c) shows a representation of this image. The third real dataset is a portion of 67 × 80 pixels cropped from the well-known Washington DC Mall hyperspectral dataset, obtained through Purdue University's MultiSpec freeware project. This dataset was acquired by the airborne mounted Hyperspectral Digital Imagery Collection Experiment (HYDICE), which measures pixel response in the 0.4 to 2.4 μm region of the visible and infrared spectrum. Bands in the 0.9 and 1.4 μm region where the atmosphere is opaque have been omitted from the dataset, leaving 191 bands. Fig. 3(d) shows an RGB representation of the area chosen for this analysis, which mainly consists of trees and a river and where the anomalous pixels belong to a building placed among the trees. The number of anomalous pixels is 15, 0.28% of the image scene.

Finally, the last real HSI was captured by one of our hyperspectral sensors mounted on an UAV. In particular, the sensor is the Specim hyperspectral FX10 which operates in the VNIR, i.e., between 0.4 and 1.0 μm. It is a push-broom camera which provides 1024 spatial pixels and 224 spectral bands [22]. However, due to the low signal-to-noise ratio of the first and last spectral bands, they were removed (1–10, 191–224), so that 180 bands have been retained. The original image scene covers an area of 1024 × 4967 pixels, but a much shorter scene has been cropped containing the anomalies of the scene. Hence, the final size of the image utilized in this work is 250 × 250 pixels as shown in Fig. 3(e). Finally, it should be mentioned that the anomaly target is a human being and that, in the whole scene, anomalies consist of 121 pixels, 0.19% of the total image, while the background represented in the scene consists of a few rows of wine crop and the soil in between.

B. Reference Algorithms

The LbL-AD algorithm has been compared against the most relevant algorithms of the state-of-the-art in the field of hyperspectral AD, both computing the image as a whole and in a line-by-line fashion as the LbL-AD algorithm does. The intention is to test the quality of the detection results and the separability between anomalies and background, as well as the complexity and required computational effort. The selected algorithms are the RX after OSPRX, which computes the anomalies once the entire image is captured, and the progressive line processing of kernel RX AD algorithm for hyperspectral imagery PLP-KRXD [31], which follows a line-by-line fashion.

C. Assessment Metrics

Receiver operating characteristic (ROC) curves and the area under these curves (AUC) have been widely used in the literature to evaluate different AD algorithms. ROC curves are two-dimensional graphical plots which illustrate the relation between true positive rates and false positive rates obtained for various threshold settings. To compare the performance of several AD algorithms, AUC is used as a scalar measure, so that a representation with the biggest AUC outperforms the others. However, this metric does not always reflect how well the algorithm separates the anomalies from the background. For this reason, two extra quality metrics will be employed in this work: Brier score (BS) and squared error ratio (SER) [32].

BS measures the accuracy of probability predictions in terms of marking anomalies and background pixels with the highest and the lowest scores, respectively. If anomaly pixels are represented as $ones$ and background pixels as $zeros$ in the ground truth, then, the BS for each type of pixel is calculated as

$$\mathrm{BS}_{\mathrm{anomaly}} = (\mathbf{p}_i - 1)^2 \tag{1}$$

$$\mathrm{BS}_{\mathrm{background}} = (\mathbf{p}_i - 0)^2 \tag{2}$$

$\mathbf{p}_i$ being the output of the algorithm, spanned in the range [0–1].

Adding the $\mathrm{BS}_{\mathrm{anomaly}}$ of all individual pixels in the image provides an insight into the error made scoring anomalous pixels, the same way that adding $\mathrm{BS}_{\mathrm{background}}$ of all individual pixels in the image gives an insight into the background misclassification error. These error scores are represented by the Anomaly_Error and Bck_Error metrics defined as follows:

$$\text{Anomaly\_Error} = \sum_{i=1}^{n_{\mathrm{anomaly(GT)}}} (\mathbf{p}_i - 1)^2 \tag{3}$$

$$\text{Bck\_Error} = \sum_{j=1}^{n_{\mathrm{bck(GT)}}} (\mathbf{p}_j - 0)^2 \tag{4}$$

Finally, a global scalar metric combining the Anomaly_Error and the Bck_Error, named SER, is obtained dividing the sum of all squared differences by the total number of pixels in the image

$$\mathrm{SER} = \frac{\sum_{i=1}^{n_{\mathrm{anomaly(GT)}}} (\mathbf{p}_i - 1)^2 + \sum_{j=1}^{n_{\mathrm{bck(GT)}}} (\mathbf{p}_j - 0)^2}{N_p} \cdot 100 \tag{5}$$

with the first addend in the summation being the Anomaly_Error and the second addend the Bck_Error.

IV. Experimental Results

A. Analysis of Input Parameters

There are a few input parameters to the algorithm, such as n, d, and *tol*, which were selected based on the experience acquired during the development of the proposed algorithm. These parameters have been adjusted after a deep analysis of the algorithm behavior in order to ensure a good performance in all analyzed scenarios. To verify that they are adequate settings, experiments have been made changing the values of each parameter in a certain range, while keeping the others constant in order to evaluate the performance of the algorithm.

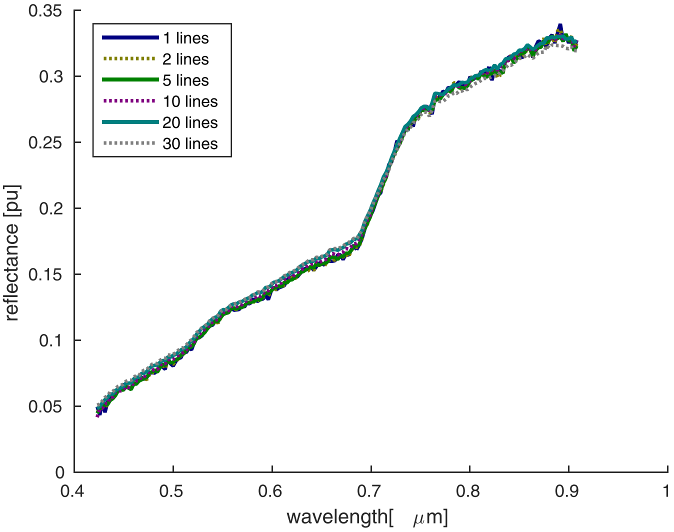


Fig. 4. Image average pixel. Analysis based on the used lines in the buffer.

As it was previously described, n is the number of captured lines that have to be accumulated in memory to calculate its average value and use it as a reference throughout the whole process. These lines are then, after captured, processed as a whole bunch and the result is provided to the user. From then on, lines are processed as they are being captured. Therefore this parameter has been set trying to achieve a good tradeoff between having a representative average value, but also small enough so the user does not have to wait a significant amount of time in order to get the detection results.

For the purpose of the analysis of this parameter, a real HSI has been selected, namely the image captured with the Specim FX10 sensor mounted on the UAV. This parameter has been progressively increased from 1 line up to 30 lines, and the average pixel vector has been represented in Fig. 4. From this figure, it is noticed that the average shows a very regular pattern independent of the value of n, being the curve smoother with the increase in the number of samples considered to compute the average. Due to this reason, a value of ten lines has been selected, as a tradeoff between enough smoothness in the curve together with a fast response to the user.

The number of representative principal components d is a rather critical parameter that has to be carefully selected. In this case, simulations show an absolute maximum value of five as a good choice in terms of information being retained and computation time. In addition, it has been imposed that the decay between two consecutive eigenvalues is not higher than 10^{-6}, otherwise, the number of elements until that decay are taken. The main reason for such restriction is that if components with a very low value are taken the algorithm is not robust enough.

Finally for the stop condition in Algorithm 3, again a tradeoff between computation time and goodness of the results has been taken into consideration. Making the tolerance for the power

TABLE I
PLP-KRXD Parameter Table

	l	w	d
Synthetic Image	15	5	1
WASP RIT	15	5	5
AVIRIS WTC	15	7	1
HYDICE WDC	15	7	2
UAV	18	6	1

TABLE II
Assessment Metric Summary

	AUC	Anomaly_Error	Bck_Error	SER
	Synthetic Image			
OSPRX	1.00	48.70	0.15	0.21
LbL-AD	1.00	77.84	0.01	0.34
PLP-KRXD	0.95	92.80	0.63	0.41
	WASP RIT			
OSPRX	0.99	40.93	1.15	0.12
LbL-AD	0.99	43.98	2.05	0.14
PLP-KRXD	0.77	71.80	10.32	0.25
	AVIRIS WTC			
OSPRX	0.97	72.93	0.76	0.18
LbL-AD	0.96	74.63	1.32	0.18
PLP-KRXD	0.51	82.97	5.87	0.22
	HYDICE WDC			
OSPRX	0.99	7.42	0.27	0.14
LbL-AD	0.99	5.69	0.48	0.11
PLP-KRXD	0.97	8.91	0.21	0.17
	UAV			
OSPRX	0.99	130.30	7.75	0.40
LbL-AD	0.99	163.48	4.46	0.49
PLP-KRXD	0.79	322.86	4.76	0.96

iteration algorithm too small has the disadvantage of having to call the function several extra times to refine the result, which does not produce an improvement in terms of the algorithm result. For this reason, a value of 10^{-8} has been selected.

B. Detection Performance

In order to test the performance of the proposed algorithm, its detection efficiency has been compared with the state-of-the-art AD algorithms presented in Section III. Due to the fact that the performance of the PLP-KRXD algorithm depends on a few input factors, such as window length l, window width w, and kernel order d, the values providing the best results for each image scene has been found out and detailed in Table I.

The results obtained in terms of the metrics described in Section III-C are summarized in Table II for each of the already mentioned images. From these results, it is concluded that the proposed LbL-AD algorithm is able to provide very similar results with respect to the OSPRX algorithm and even outperforms it in some cases, as it is in the case of the HYDICE WDC image, with the major advantage of being able to compute the anomalies present in the image as soon as the line of pixels is sensed. In addition, it is also concluded from Table II that if we focus our attention only on the solution that allows to process the HSIs in a line-by-line fashion, our proposal delivers a much better detection performance than the one provided by the PLP-KRXD.

Finally, Fig. 5 shows the two-dimensional plots of the detection results given by the proposed LbL-AD algorithm and the OSPRX algorithm for all the considered datasets together with the ground truth. It is visually demonstrated that our proposal is able to capture in a very precise way all the anomalies present in a given HSI independently of the characteristics of the background and of the anomalies present in the scene under analysis, and also showing that the visual result provided by the LbL-AD algorithm is very similar to the one provided by the OSPRX as it was already demonstrated with the results shown in Table II.

C. Execution Performance

In Section IV-B, the detection performance comparison of the LbL-AD algorithm versus the state-of-the-art has revealed that the results obtained by our proposal are much better than the ones provided by the line-by-line PLP-KRXD algorithm and equivalent to those provided by the OSPRX algorithm. The main advantage of the proposed LbL-AD algorithm with respect to the OSPRX algorithm is its capability of processing the lines of pixels as they are being sensed, while the OSPRX needs to wait until the complete image has been captured to produce a result. In order to emphasize this aspect, a comparison between the time required for the UAV image to be processed and have the results ready for both algorithms is shown in Fig. 6.

As it can be seen, in case the image is processed with the OSPRX, no matter where the anomaly is located, information will be available after 1.77 s. If the image is processed with the proposed LbL-AD algorithm, depending on where the anomaly is present at the image, a result will be available for the system. For instance if the anomaly is located at a 40% of the image, that is, in this case, at line 100, the result is available after approximately 0.7 s, which is more than a second earlier than with the other method.

The test has been performed by setting the parameter n to 10, the total number of lines N to 250 and the sampling rate at which the image was captured to 150 lines per second. The algorithms were executed on a PC with an Intel Core i7-6700 CPU

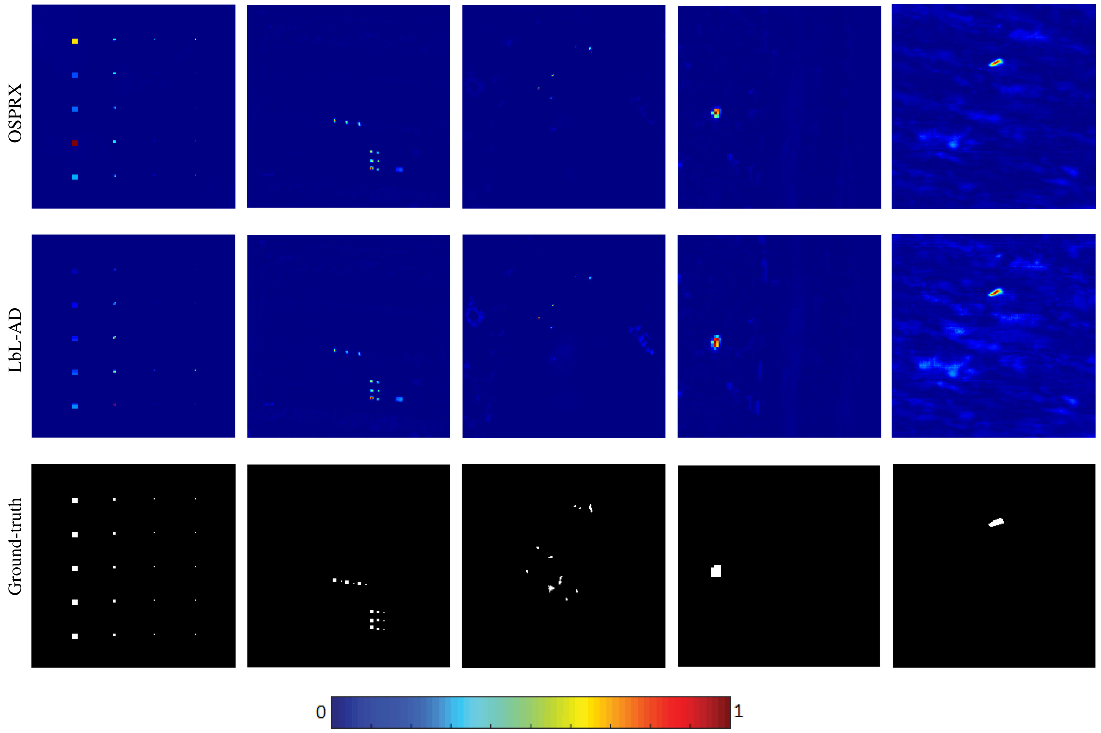


Fig. 5. Two-dimensional plots of the detection results obtained by the AD algorithms considered in this work (rows) for all datasets (columns). To make the results easier to understand, the ground truths of the anomalous maps are shown in the last row.

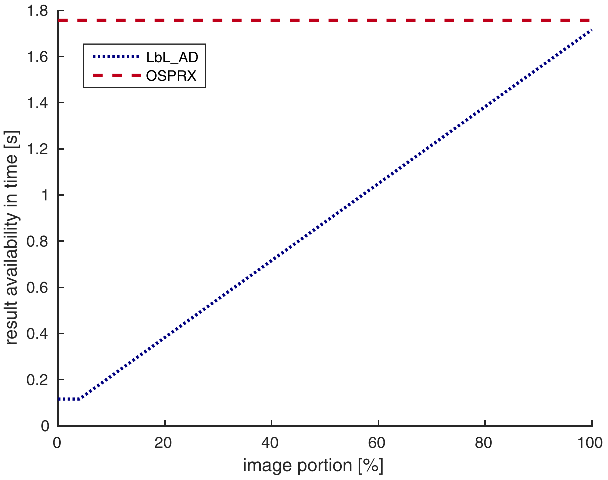


Fig. 6. Execution time comparison in terms of availability of the result.

and 16 GB of RAM memory running a Ubuntu 14.04 operating system.

V. Conclusion and Further Research

This work describes a novel AD technique (LbL-AD) for push-broom hyperspectral sensors. The algorithm is premised on the concept of orthogonal subspace projection, similar to other ones encountered in the scientific literature, but after an initial stage involving the first n scanned lines, it is combined with a line-by-line processing chain. A detailed behavior of the algorithm is described and experimental results are obtained using quality metrics for different synthetic and real images as, for instance, the one captured with our hyperspectral flying platform, showing an excellent performance compared to the relevant state-of-the-art proposals, such as OSPRX and PLP-KRXD. From these results, it is concluded that LbL-AD is an attractive candidate for hyperspectral imagery, opening a new frontier in a myriad of applications in which real time and low computational cost is mandatory.

References

[1] J. A. Berni, P. J. Zarco-Tejada, M. D. Suárez Barranco, and E. Fereres Castiel, “Thermal and narrow-band multispectral remote sensing for vegetation monitoring from an unmanned aerial vehicle,” *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 3, pp. 722–738, Mar. 2009.

[2] T. Adão *et al.*, “Hyperspectral imaging: A review on UAV-based sensors, data processing and applications for agriculture and forestry,” *Remote Sens.*, vol. 9, no. 11, 2017, Art. no. 1110.

[3] S. J. Kim and G. J. Lim, “A hybrid battery charging approach for drone-aided border surveillance scheduling,” *Drones*, vol. 2, no. 4, 2018, Art. no. 38. [Online]. Available: http://www.mdpi.com/2504-446X/2/4/38

[4] M. Adamski, “Analysis of propulsion systems of unmanned aerial vehicles,” *J. Mar. Eng. Technol.*, vol. 16, no. 4, pp. 291–297, 2017. [Online]. Available: https://doi.org/10.1080/20464177.2017.1383337

[5] O. González-Espasandín, T. J. Leo, and E. Navarro-Arévalo, “Fuel cells: A real option for unmanned aerial vehicles propulsion,” *Scientific World J.*, vol. 2014, 2014, Art. no. 497642. [Online]. Available: https://doi.org/10.1155/2014/497642

[6] W. Lu, D. Zhang, J. Zhang, T. Li, and T. Hu, “Design and implementation of a gasoline-electric hybrid propulsion system for a micro triple tilt-rotor VTOL UAV,” in *Proc. 6th Data Driven Control Learn. Syst.*, May 2017, pp. 433–438.

[7] L. Merino, F. Caballero, J. R. Martínez-de Dios, I. Maza, and A. Ollero, "An unmanned aircraft system for automatic forest fire monitoring and measurement," *J. Intell. Robot. Syst.*, vol. 65, no. 1, pp. 533–548, Jan. 2012. [Online]. Available: https://doi.org/10.1007/s10846-011-9560-x

[8] D. Kim, M. Liu, S. Lee, and V. R. Kamat, "Remote proximity monitoring between mobile construction resources using camera-mounted UAVs," *Autom. Construction*, vol. 99, pp. 168–182, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0926580518304102

[9] V. Baiocchi, D. Dominici, and M. Mormile, "UAV application in post-seismic environment," in *Proc. Int. Arch. Photogrammetry, Remote Sens. Spatial Inf. Sci.*, 2013, pp. 21–25.

[10] J. Sun, B. Li, Y. Jiang, and C.-Y. Wen, "A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes," *Sensors*, vol. 16, no. 11, 2016, Art. no. 1778. [Online]. Available: http://www.mdpi.com/1424-8220/16/11/1778

[11] Y. Lin, J. Hyyppa, and A. Jaakkola, "Mini-UAV-borne LIDAR for fine-scale mapping," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 426–430, May 2011.

[12] S. Esposito *et al.*, "Performance evaluation of lightweight lidar for UAV applications," in *Proc. IEEE Geosci. Remote Sens. Symp.*, Jul. 2014, pp. 792–795.

[13] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990.

[14] A. Schaum, "Joint subspace detection of hyperspectral targets," in *Proc. IEEE Aerosp. Conf.*, vol. 3, 2004.

[15] C.-I. Chang, "Orthogonal subspace projection (osp) revisited: A comprehensive study and analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 502–518, Mar. 2005.

[16] L. Zhu, G. Wen, S. Qiu, and X. Zhang, "Improving hyperspectral anomaly detection with a simple weighting strategy," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 95–99, Jan. 2019.

[17] M. Vafadar and H. Ghassemian, "Hyperspectral anomaly detection using combined similarity criteria," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4076–4085, Nov. 2018.

[18] H. Su, Z. Wu, Q. Du, and P. Du, "Hyperspectral anomaly detection using collaborative representation with outlier removal," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 12, pp. 5029–5038, Dec. 2018.

[19] N. Ma, Y. Peng, and S. Wang, "A fast recursive collaboration representation anomaly detector for hyperspectral image," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 588–592, Apr. 2019.

[20] N. Huyan, X. Zhang, H. Zhou, and L. Jiao, "Hyperspectral anomaly detection via background and potential anomaly dictionaries construction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 4, pp. 2263–2276, Apr. 2019.

[21] Y. Xu, Z. Wu, J. Chanussot, and Z. Wei, "Joint reconstruction and anomaly detection from compressive hyperspectral images using Mahalanobis distance-regularized tensor RPCA," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2919–2930, May 2018.

[22] Specim, FX series hyperspectral cameras. [Online]. Available: http://www.specim.fi/fx/. Accessed on: Jun. 11, 2019.

[23] DJI Matrice 600. [Online]. Available: https://www.dji.com/matrice600. Accessed on: Jun. 11, 2019.

[24] C. Lanczos, "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators," *J. Res. Nat. Bur. Stand. B*, vol. 45, pp. 255–282, 1950.

[25] R. B. Lehoucq and D. C. Sorensen, "Deflation techniques for an implicitly restarted Arnoldi iteration," *SIAM J. Matrix Anal. Appl.*, vol. 17, no. 4, pp. 789–821, 1996.

[26] Power method tutorial, MATLAB central file exchange, 2007. [Online]. Available: http://web.mit.edu/18.06/www/Spring17/Power-Method.pdf. Accessed on: Jun. 11, 2019.

[27] USGS digital spectral library. [Online]. Available: http://speclab.cr.usgs.gov/spectral-lib.html. Accessed on: Jun. 11, 2019.

[28] Hyperspectral Imagery Synthesis Toolbox, Grupo de Inteligencia Computacional, Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU), Spain. [Online]. Available: http://www.ehu.es/ccwintco/index.php/Hyperspectral_Imagery_Synthesis_tools_for_MATLAB. Accessed on: Jun. 11, 2019.

[29] J. A. Herweg *et al.*, "Spectir hyperspectral airborne rochester experiment data collection campaign," in *Proc. SPIE Defense, Secur., Sens.*, 2012, pp. 839028–839028.

[30] A. Plaza, Q. Du, Y.-L. Chang, and R. L. King, "High performance computing for hyperspectral remote sensing," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 4, no. 3, pp. 528–544, 2011.

[31] C. Zhao, W. Deng, Y. Yan, and X. Yao, "Progressive line processing of kernel RX anomaly detection algorithm for hyperspectral imagery," *Sensors*, vol. 17, no. 8, 2017, Art. no. 1815. [Online]. Available: http://www.mdpi.com/1424-8220/17/8/1815

[32] M. Diaz, R. Guerra, S. Lopez, and R. Sarmiento, "An algorithm for an accurate detection of anomalies in hyperspectral images with a low computational complexity," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1159–1176, Feb. 2018.

Pablo Horstrand was born in Las Palmas de Gran Canaria, Spain, in 1986. He received the double degree in industrial engineering and electronics and control engineering and the master's degree in telecommunication technologies from the University of Las Palmas de Gran Canaria (ULPGC), Las Palmas de Gran Canaria, Spain, in 2010 and 2011, respectively. He is currently working toward the Ph.D. degree in telecomunication technologies at the ULPGC.

He worked with ABB Switzerland, Aargau, Switzerland, first in the Minerals and Printing, Drives Department, and last in the Traction Department, between 2011 and 2017. In 2018, he was with the Signal & Image Centre Department, Royal Military Academy, Belgium, for his Ph.D. research.

María Díaz was born in Spain in 1990. She received the industrial engineer degree from the University of Las Palmas de Gran Canaria, Spain, in 2014, and the master's degree in system and control engineering jointly from the Universidad Complutense de Madrid, Madrid, Spain, and the Universidad Nacional de Educación a Distancia (UNED), Madrid, Spain, in 2017. She is currently working toward the Ph.D. degree with the University of Las Palmas de Gran Canaria, developing her research activities at the Integrated Systems Design Division of the Institute for Applied Microelectronics (IUMA).

In 2017, she conducted a research stay with the GIPSA-lab, University of Grenoble Alpes, France. Her research interests include image and video processing, development of highly parallelized algorithms for hyperspectral images processing and hardware implementation.

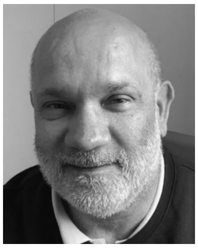
Raúl Guerra was born in Las Palmas de Gran Canaria, Spain, in 1988. He received the industrial engineer degree from the University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, in 2012, the master's degree in telecommunications technologies from the Institute of Applied Microelectronics, Las Palmas de Gran Canaria, in 2013, and the Ph.D. degree in telecommunications technologies from the University of Las Palmas de Gran Canarias in 2017.

In 2016, he worked as a Researcher with the Configurable Computing Lab in Virginia Tech University, Blacksburg, VA, USA. His research interests include the parallelization of algorithms for multispectral and hyperspectral images processing and hardware implementation.

Sebastián López (M'08–SM'15) was born in Las Palmas de Gran Canaria, Spain, in 1978. He received the electronic engineering degree from the University of La Laguna, San Cristobal de La Laguna, Spain, in 2001, and the Ph.D. degree in electronic engineering from the University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, Spain, in 2006.

He is currently an Associate Professor with the University of Las Palmas de Gran Canaria, where he is involved in research activities with the Integrated Systems Design Division, Institute for Applied Microelectronics. He has coauthored more than 120 papers in international journals and conferences. His research interests include real-time hyperspectral imaging, reconfigurable architectures, high-performance computing systems, and image and video processing and compression.

Dr. López was the recipient of regional and national awards during his electronic engineering degree. He is an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, *MDPI Remote Sensing*, and the *Mathematical Problems in Engineering Journal*. He was an Associate Editor for the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS from 2008 to 2013. He also serves as an active Reviewer for different JCR journals and as program committee member of a variety of reputed international conferences. Furthermore, he acted as one of the Program Chairs of the IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS) in its 2014 edition, and of the SPIE Conference of High Performance Computing in Remote Sensing from 2015 to 2018. Moreover, he has been the Guest Editor of different special issues in JCR journals related with his research interests.



José F. López received the M.S. degree in physics (specializing in electronics) from the University of Seville, Sevilla, Spain, in 1989, and the Ph.D. degree from the University of Las Palmas de Gran Canaria (ULPGC), Las Palmas de Gran Canaria, Spain, in 1994, being awarded by this University for his research in the field of High Speed Integrated Systems.

He was with Thomson Composants Microondes, Orsay, France, in 1992. In 1995, he was with the Center for Broadband Telecommunications at the Technical University of Denmark (DTU), Lyngby, Denmark, and in 1996, 1997, 1999, and 2000, he was Visiting Researcher with the Edith Cowan University (ECU), Perth, Western Australia. He has conducted his investigations at the Institute for Applied Microelectronics (IUMA), where he has been acting as Deputy Director since 2009. He is currently a Lecturer with the School of Telecommunication & Electronics Engineering and with the M.Sc. Program of IUMA, in ULPGC. He has authored/coauthored around 140 papers in national and international journals and conferences. His current research interests include image processing, UAVs, hyperspectral technology, and their applications.

Dr. López has been actively enrolled in more than 40 research projects funded by the European Community, Spanish Government, and international private industries located in Europe, USA, and Australia.