Adaptation of the CCSDS 123.0-B-2 Standard for RGB Video Compression

Yubal Barrios, Raúl Guerra, Sebastián López, *Senior Member, IEEE*, and Roberto Sarmiento

***Abstract*—The integration of video sensors on-board satellites is becoming a trend in the space industry, since they provide extra information in the temporal domain when compared with traditional remote sensing imaging acquisition equipment. The inclusion of the temporal dimension together with the constant increase in the sensor resolution supposes a challenge for on-board processing, taking into account the limited computational and storage resources on-board satellites and that it is unfeasible to directly transmit raw video to ground, due to downlink bandwidth limitations. For these reasons, on-board video compression is needed. However, the inherent complexity of the video encoders used on ground limits their implementation on environments with high constraints in terms of computational burden, area, and power consumption. This article proposes an extended compression chain that implements as compression core the CCSDS 123.0-B-2 standard, originally developed for near-lossless compression of multi- and hyperspectral images. In addition, some preprocessing stages are included to manage the temporal dimension of RGB videos efficiently. The proposed solution guarantees low complexity and flexibility to compress both multi- and hyperspectral images and panchromatic and RGB videos by using a single compression instance, which is adapted by adding or removing the appropriate stages. Results demonstrate the viability of this solution to be implemented on space payloads, since high compression ratios are achieved without incurring in a penalty in terms of video quality.**

***Index Terms*—Consultative committee for space data systems (CCSDS), multispectral imaging, on-board processing, space missions, video compression.**

I. Introduction

VIDEO sensors are gaining interest, in order to be embarked on satellites, as remote sensing instruments that provide information not only in the spatial dimension but also in the temporal domain, capturing data continuously that allow new on-board applications, such as disaster monitoring or target detection and tracking in real time [1]. Although the spatial resolution is decreased to compensate the volume of information generated by considering the temporal domain, raw data volume to be managed, stored, or directly transmitted to ground is still high. This fact becomes mandatory for the implementation of video compression solutions on-board satellites. This need will be even more critical in the next few years with the emerging multispectral video sensors [2], [3], which also add information in the spectral domain at different wavelengths, together with an increased pixel resolution, needing of efficient 4-D compression techniques (i.e., considering spatial, spectral, and temporal dimensions) that allow us to reduce the acquired video size without incurring in a penalty in terms of video quality [4], [5].

Targeting RGB and multispectral on-board video compression, a tailored version of commercial video encoder widely used on ground applications, such as the H.264/AVC specification [6], could be adapted to work on-board satellites, as it is recommended by the space agencies for real-time applications with data transmissions up to 25 Mb/s [7]. Although this option offers good performance in terms of compression rate-distortion ratio, it also presents coarse drawbacks in order to be implemented on hardware, such as a complex architecture (especially the interprediction stage, where motion estimation is computed), preventing its implementation on hardware resources available on-board satellites [8], or an imprecise behavior for lossless compression, among others. Different works are available in the state of the art about field-programmable gate array (FPGA) implementations of the H.264 encoder, but focusing, in particular, on stages whose performance is critical, such as motion estimation [9]–[12], intraprediction [13], [14], quantization [15], or encoding [16], [17]. A full hardware implementation of the H.264 encoder in the baseline profile is presented in [18], consuming 89% of slices available in a Xilinx XC6VLX240T FPGA.

In addition, as commercial video encoders are usually based on a transform-based approach, they yield higher compression ratios than prediction-based methods. This is achieved by introducing high errors in the frame areas with higher entropy without incurring in a video degradation by the human perception. However, it is extremely difficult to accurately control the error introduced in this kind of encoders, being them not interesting for specific applications that require near-lossless compression. Alternative compression methods based on content-weighted models have been recently proposed to take into consideration the spatial variation presented in an image [19], [20]. Although results in terms of decompressed image quality are promising, adaptations for a video acquisition scenario are required to also take into account temporal information; at the same time, the complexity should be reduced to be embarked on satellites.

Due to all these reasons, a different strategy has been followed in the presented approach, which focuses on the development

Manuscript received October 13, 2021; revised December 23, 2021; accepted January 20, 2022. Date of publication January 25, 2022; date of current version February 16, 2022. This work was supported by the European Union's Horizon 2020 Research and Innovation Program through the Video Imaging Demonstrator for Earth Observation project under Grant 870485. *(Corresponding author: Sebastián López.)*

The authors are with the Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain (e-mail: ybarrios@iuma.ulpgc.es; rguerra@iuma.ulpgc.es; seblopez@iuma.ulpgc.es; roberto@iuma.ulpgc.es).

Digital Object Identifier 10.1109/JSTARS.2022.3145751

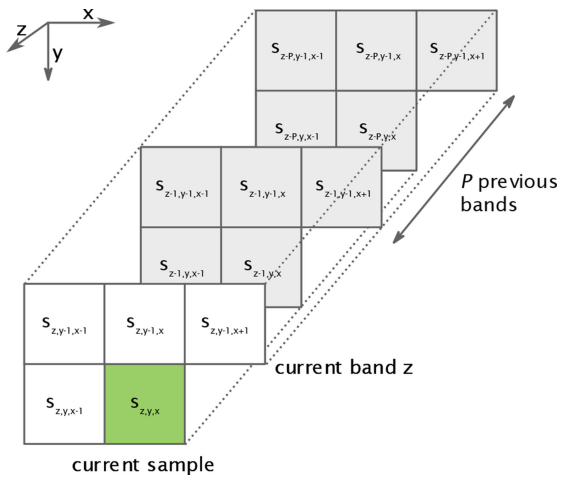


Fig. 1. Spatial and spectral vicinity used during the prediction.

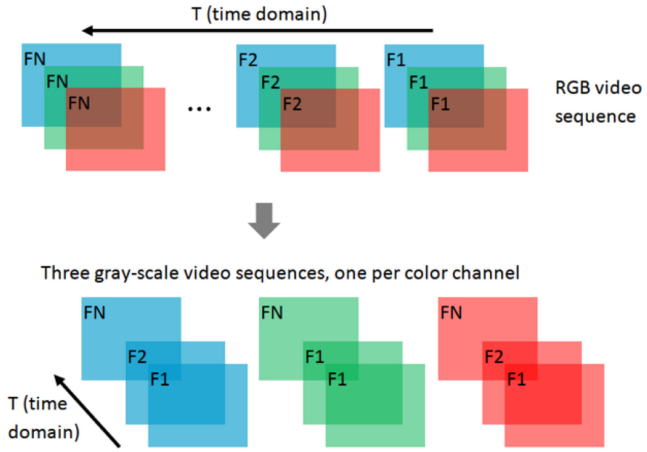


Fig. 2. Compression strategy for an RGB video, treating each color channel as independent frames.

of a new video compression algorithm specifically designed for being efficiently executed on-board satellites. This solution is able to deal with all the requirements imposed by remote sensing applications, such as both an acceptable compression ratio and the video quality after reconstruction, without incurring in a penalty in terms of architectural complexity and power consumption. In particular, a low-complexity solution for RGB video compression on-board satellites is presented in this article. The proposed solution uses as compression core the Consultative Committee for Space Data Systems (CCSDS) 123.0-B-2 multi- and hyperspectral image compression standard [21], also providing the capability to compress 3-D images by using the same processing core just removing the unnecessary preprocessing stages. Compression of panchromatic or grayscale video is also guaranteed by considering it as an RGB video with only one band. Since the CCSDS 123.0-B-2 compression standard was specifically developed to fit well on hardware resources available on-board satellites, we ensure that the proposed approach presents a low complexity and a reduced power consumption, without incurring in a penalty neither in the compression ratio nor the reconstructed video quality. In addition, losses introduced in the compression chain can be controlled in a band-by-band basis, allowing to compress spectral channels with different error values, which is an strength compared to transform-based video encoders.

The rest of this article is organized as follows. Section II explains the compression core of the proposed solution, based on the CCSDS 123.0-B-2 standard, distinguishing between the prediction and the entropy coding stages. Then, Section III describes the compression chain in detail, highlighting the new preprocessing stages to manage the temporal dimension of the RGB video. Next, experimental results are presented in Section IV, also explaining the used dataset. Finally, Section V concludes this article.

II. CCSDS 123.0-B-2 Standard

The CCSDS, an international entity comprised by the main space agencies and responsible of developing data compression standards for space applications taking into account limitations in terms of computational burden, area occupation, and power consumption, has recently published the CCSDS 123.0-B-2 standard [21]. The proposed algorithm focuses on the near-lossless compression of multi- and hyperspectral images, and it is comprised of two main stages: a predictive-based approach for spectral and spatial decorrelation and an entropy coder, whose main purpose is to represent the input prediction residuals with the minimum possible number of bits but ensuring at the same time a proper decompression.

A. Prediction Stage

The prediction-based preprocessor defined in the CCSDS 123.0-B-2 compression standard is able to estimate the value of the current input sample $s_{z,y,x}$ taking into account previously processed samples in its spatial and spectral vicinity, as shown in Fig. 1. In this sense, a critical parameter is the number of previous bands P used to compute the prediction, which is a user-defined parameter in the range $0 \leq P \leq 15$. A value of $P = 0$ supposes that only spatial information is considered to perform the prediction (i.e., the spectral correlation is not exploited), while values higher than 3 do not considerably improved prediction capabilities.

The first step is to perform a local sum $\sigma_{z,y,x}$, which is a weighted sum of neighboring samples in the current band z. The vicinity considered in this calculation depends on the selected local sum, having two main options: 1) neighbor-oriented, which uses all the adjacent samples to compute the local sum, including the ones located at the left, top-left, top, and top-right positions of the current one, and 2) column-oriented, which only takes into account the samples at the left and the top of the current one to compute the local sum, with a higher weight that the one assigned to these samples in the neighbor-oriented option.

In addition, the CCSDS 123.0-B-2 standard introduces a novelty compared to its predecessor, which is the possibility of using *narrow* local sums. This feature, applicable to both neighbor- and column-oriented local sums, does not use the sample at the left of the current one in the same band z for

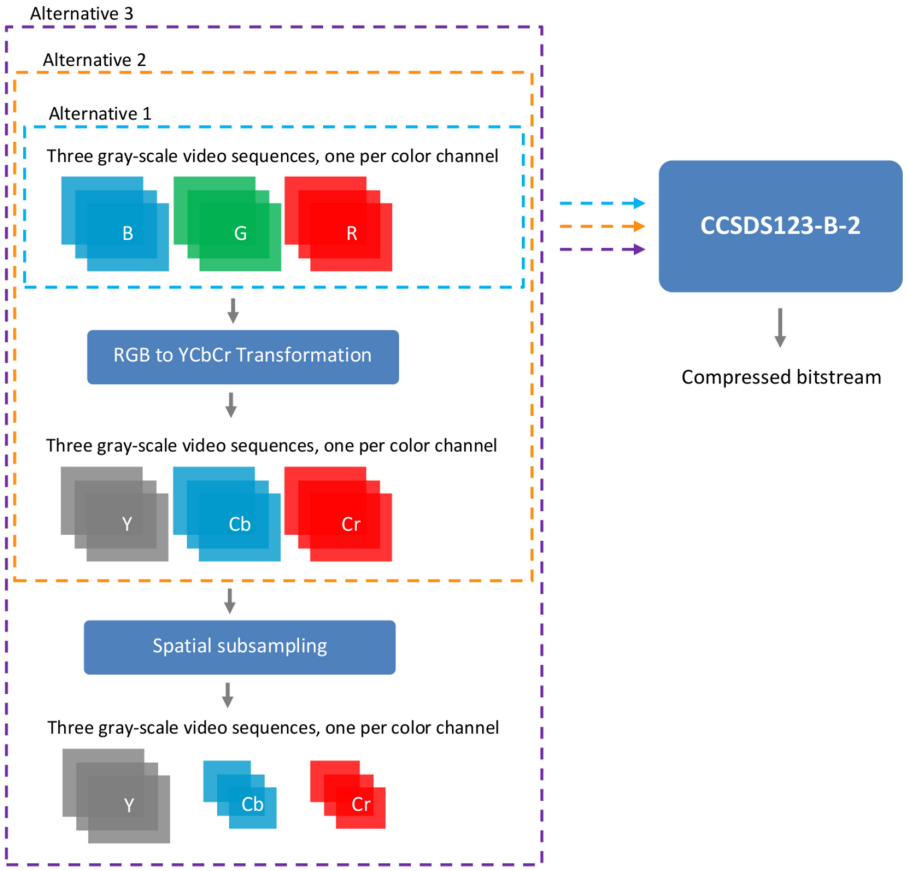


Fig. 3. Full processing chain for RGB video compression.

the computation (it is replaced by the sample at the left in the previous band $s_{z-1,y,x-1}$), improving throughput on hardware implementations by removing data dependencies in the process of consecutive input samples. Equations (1) and (2) indicate how to calculate the local sums under the wide and narrow neighbor-oriented modes, respectively. In a similar way, (3) and (4) reflect the computation of the wide and narrow column-oriented local sums, respectively. It is also remarkable that sample representative values $s''_{z,y,x}$, which will be explained later, are used to compute the local sums, under near-lossless compression; if lossless compression is selected, input samples are directly employed

$$\sigma_{z,y,x}=\begin{cases} s''_{z,y,x-1}+s''_{z,y-1,x-1}+s''_{z,y-1,x}+s''_{z,y-1,x+1}, & y>0,0<x<N_x-1 \\ 4s''_{z,y,x-1}, & y=0,x>0 \\ 2(s''_{z,y-1,x}+s''_{z,y-1,x+1}), & y>0,x=0 \\ s''_{z,y,x-1}+s''_{z,y-1,x-1}+2s''_{z,y-1,x}, & y>0,x=N_x-1 \end{cases} \tag{1}$$

$$\sigma_{z,y,x}=\begin{cases} s''_{z,y-1,x-1}+2s''_{z,y-1,x}+s''_{z,y-1,x+1}, & y>0,0<x<N_x-1 \\ 4s''_{z-1,y,x-1}, & y=0,x>0,z>0 \\ 2(s''_{z,y-1,x}+s''_{z,y-1,x+1}), & y>0,x=0 \\ 2(s''_{z,y-1,x-1}+s''_{z,y-1,x}), & y>0,x=N_x-1 \\ 4s_{\text{mid}}, & y=0,x>0,z=0 \end{cases} \tag{2}$$

$$\sigma_{z,y,x}=\begin{cases} 4s''_{z,y-1,x}, & y>0 \\ 4s''_{z,y,x-1}, & y=0,x>0 \end{cases} \tag{3}$$

$$\sigma_{z,y,x}=\begin{cases} 4s''_{z,y-1,x}, & y>0 \\ 4s''_{z-1,y,x-1}, & y=0,x>0,z>0 \\ 4s_{\text{mid}}, & y=0,x>0,z=0 \end{cases}. \tag{4}$$

The next step is to calculate the local differences, which makes use of the local sum value. Two possible prediction modes are available, and the number of differences taken into account depends on the selected option: the *reduced* mode just uses the central differences in the P previous bands, which are calculated as $d_{z,y,x}=4s''_{z,y,x}-\sigma_{z,y,x}$, considering only samples in the same position as the current one, but in previously processed spectral bands; on the other side, the *full* mode considers the directional differences, in addition to the central ones. These directional differences are computed, as shown in (5)–(7), employing the adjacent samples in that specific direction in the current band z, with their associated weight. Once the local differences have been calculated, depending on the selected prediction mode, the local differences vector $U_{z,y,x}$ is formed. Then, an inner product is performed, element by element, between this vector and a weight vector, $W_{z,y,x}$, maintaining an independent weight vector per band z. Weight elements are updated with each new sample, adapting the predictor performance by taking into account statistics of previously processed pixels

$$d^N_{z,y,x}=\begin{cases} 4s''_{z,y-1,x}-\sigma_{z,y,x}, & y>0 \\ 0, & x>0,y=0 \end{cases} \tag{5}$$

$$d^{W}_{z,y,x} = \begin{cases} 4s''_{z,y,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s''_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & x > 0, y = 0 \end{cases} \quad (6)$$

$$d^{NW}_{z,y,x} = \begin{cases} 4s''_{z,y-1,x-1} - \sigma_{z,y,x}, & x > 0, y > 0 \\ 4s''_{z,y-1,x} - \sigma_{z,y,x}, & x = 0, y > 0 \\ 0, & x > 0, y = 0 \end{cases}. \quad (7)$$

The result of this inner product is the predicted central local difference $\hat{d}_{z,y,x}$, which is used to calculate the predicted sample $\hat{s}_{z,y,x}$, in addition to some user-defined parameters. According to [22], the calculation of the predicted sample can be simplified as

$$\hat{s}_{z,y,x} \approx \left\lfloor \frac{\hat{d}_{z,y,x} + 2^{\Omega}\sigma_{z,y,x}}{2^{\Omega+2}} \right\rfloor. \quad (8)$$

The prediction residual $\Delta_z(t)$, which is the difference between the predicted and the current sample, is sent to the quantizer when near-lossless compression is selected or it is directly mapped into an unsigned integer to conform the mapped residual $\delta_{z,y,x}$, which is transmitted to the entropy coder under lossless compression.

The quantizer makes use of a uniform bin size $2m_z(t)+1$ to introduce losses during the predictor, with $m_z(t)$ being the maximum error limit. Lossless compression is guaranteed if $m_z(t) = 0$. The maximum error limit $m_z(t)$ is controlled by defining a maximum *absolute error* a_z and/or a *relative error* limit r_z, which can be the same for the whole image (i.e., band-independent) or different for each band z (i.e., band-dependent). The maximum error value is limited by its dynamic range (D_a and D_r for absolute and relative errors, respectively), which is a value in the range $1 \leq D_a, D_r \leq \min(D-1, 16)$, with D being the dynamic range (i.e., the bit precision) of the input samples. (9) summarizes how to compute the quantizer index $q_z(t)$, which is the output of the quantizer stage

$$q_z(t) = \text{sgn}(\Delta_z(t)) \left\lfloor \frac{|\,\Delta_z(t)\,| + m_z(t)}{2m_z(t)+1} \right\rfloor. \quad (9)$$

Sample representatives, introduced at the beginning of this subsection, are needed to reduce the impact of the quantization, reconstructing approximately the original samples $s_z(t)$. Three user-defined parameters are used to control the deviation of the sample representative values: the sample representative resolution Θ; the damping ϕ_z, which limits the effect of noisy samples; and the offset ψ_z, which tunes the sample representative value toward $s'_z(t)$ (i.e., the quantizer bin center) or $\hat{s}_z(t)$. The range of allowed values for both ϕ_z and ψ_z is limited between 0 and $2^{\Theta}-1$. Nonzero values for ϕ_z and ψ_z tend to provide higher compression performance if there is a spectral correlation between adjacent bands, as concluded in [23].

B. Entropy Coding

The CCSDS 123.0-B-2 compression standard defines three possible options for the entropy coding stage: the sample-adaptive, proposed in Issue 1 of the standard; the block-adaptive, initially defined in the CCSDS 121 universal lossless compression standard [24]; and the new hybrid encoder, which is part of the recent Issue 2 of the CCSDS 123 standard. This latter encoder is the one selected for the proposed compression chain, since it outperforms the other options for low bit rates, especially for low-entropy input data.

This encoder uses one of the two possible encoding techniques, named as *high-entropy* and *low-entropy*, depending on the value of (10). If the condition is met, the high-entropy mode is used; otherwise, low-entropy is applied. Code selection statistics (a counter $\Gamma(t)$ and an accumulator $\Sigma_z(t)$) are maintained per band z, and they are essential to select one encoding method or other. These statistics are updated with every new sample, and also, they are rescaled periodically. T_0 represents a threshold specified in the standard that also determines which encoding

Fig. 4. Single frame of the video sequences captured by the IDS uEye sensor.

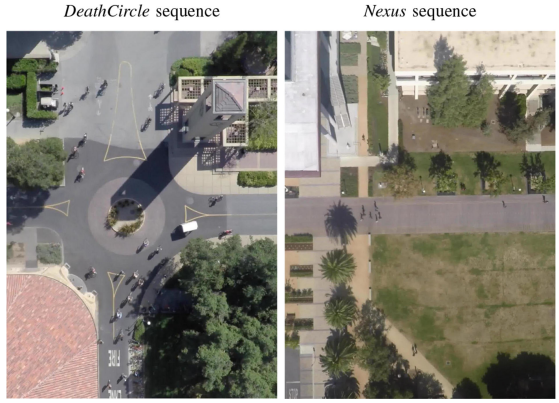


Fig. 5. Single frame of the video sequences from the Stanford Drone dataset.

TABLE I
CCSDS 123.0-B-2 CONFIGURATION USED FOR VERIFICATION

Parameter	Value	Description
Video parameters		
D	8	Bit-width of input pixels
ENDIANNESS	0	Little Endian
IS_SIGNED	0	Unsigned samples
Predictor parameters		
LOCAL_SUM	Wide neighbour-oriented	Local sum computation method
PRED_MODE	Full	Selected prediction mode
P	3	Previous frames used for prediction (by channel)
Ω	16	Weight resolution
v_{min}	0	Weight update initial value
v_{max}	3	Weight update final value
t_{inc}	64	Weight update change interval
Θ	2	Sample representatives resolution
ϕ	1	Damping
ψ	1	Offset
D_a	6	Absolute error bit-depth
A	[0:31]	Band-independent absolute error
Encoder parameters		
U_{max}	32	Unary Length Limit
$\gamma*$	4	Rescaling Counter Size
γ_0	1	Initial Count Exponent

method should be applied to the current input pixel

$$\Sigma_z(t) \cdot 2^{14} \geq T_0 \cdot \Gamma(t). \tag{10}$$

In the high-entropy mode, a single-output codeword is generated by each processed sample. It uses Reverse Length-Limited Golomb Power-of-2 codes, which are similar to the encoding procedure followed by the sample-adaptive encoder, but proceeding in reverse order. The following assumptions are taken into account.

1) If $\delta(t)/2^k < U_{\max}$, the output codeword is comprised of the k least significant bits of $\delta(t)$, followed by a "1" and $\delta(t)/2^k$ 0s. The parameter k is known as code index, while $\delta(t)$ is the mapped residual and the unary length limit $U_{\max}$ is a user-defined parameter.
2) Otherwise, the output codeword consists of the representation of the mapped residual $\delta(t)$ with *D* bits, being *D* the dynamic range of each input pixel. Finally, the codeword is completed with $U_{\max}$ zeros.

Under the low-entropy mode, one of the 16 variable-to-variable length codes is used to encode each mapped residual. The low-entropy method may encode multiple samples in a single codeword, allowing an enhancement of the compression performance. Each low-entropy code has an active prefix, which is a sequence of input symbols initialized to the null value. The selection of the appropriate active prefix depends on the value of the current $\delta(t)$, the coding statistics, and certain thresholds T_i defined by the standard, as shown in (11). The code index i, one per each of the 16 variable-to-variable codes, determines if $\delta(t)$ is appended to the selected active prefix; if the input symbol is the escape one (represented as X), the residual value $\delta_(t) - L_i - 1$ is sent to the output bitstream and coded in the same way than the high-entropy method, with a k value equal to 0. The parameter L_i is the input symbol limit for that specific code index i, which is also predefined in the standard

$$\Sigma_z(t) \cdot 2^{14} < \Gamma(t) \cdot T_i. \tag{11}$$

If after updating the corresponding active prefix, it is equal to a complete input codeword, the associated output codeword to that input sequence is appended to the output bitstream, and then, the active prefix for that low-entropy code is reset. It is also possible that after processing the last mapped residual, one or more than one of the active prefixes do not match any of the complete input codewords in their corresponding tables, in their current state. In this case, the standard defines 16 input-to-output flush codeword tables that contain all the possible combinations of incomplete input codewords for each code index i.

Finally, a compressed image tail is generated and appended at the end of the bitstream. It includes the necessary information to properly decode the compressed bitstream in reverse order, including the flush codewords, in increasing order, for each of the 16 active prefixes, and the final value of the accumulator $\Sigma_z(t)$ in each spectral band z. In addition, a header is included at the beginning of the output bitstream with the used values for each one of the hybrid encoder parameters, ensuring a correct decompression on ground.

III. SOLUTIONS FOR RGB VIDEO COMPRESSION USING THE CCSDS 123 COMPRESSION CORE

A. *Using a Single CCSDS 123 Compression Core for Each Color Channel*

When the CCSDS 123.0-B-2 algorithm is used for compressing multi- and hyperspectral images, it has been demonstrated that the higher compression rate-distortion ratios are obtained for hyperspectral images with a large number of bands [25]. This happens due to the fact that when the number of bands increases, the spectral channels at which subsequent bands are sensed get closer, and hence, adjacent bands start to be strongly correlated. Accordingly, each sample can be very accurately predicted using the values of the same pixel in previous bands. However, when compressing multispectral images with higher spectral distances between consecutive bands, the spectral prediction is not so efficient. The same situation is applicable to remote sensing applications focused on video acquisition, where a high spatial resolution results in video sequences with a reduced global movement between subsequent frames, and where the moving objects are very small in comparison to the spatial dimension of the captured scene. Due to this, the correlation between subsequent frames for the same color channel in an RGB video could be strong, while the correlation between the three color channels for a single frame would not be so high (i.e., a special case of multispectral image with just three bands). Hence, each of the color channels could be treated as an independent grayscale video and can be independently compressed using the CCSDS 123.0-B-2, as shown in Fig. 2.

Following this procedure, both panchromatic and RGB video sequences can be compressed using the same CCSDS 123.0-B-2 core, without carrying out any further modification, just by replacing the spectral dimension z by the temporal dimension t (i.e., using for the inter-prediction samples in the P previous frames, instead of the vicinity in the previously processed

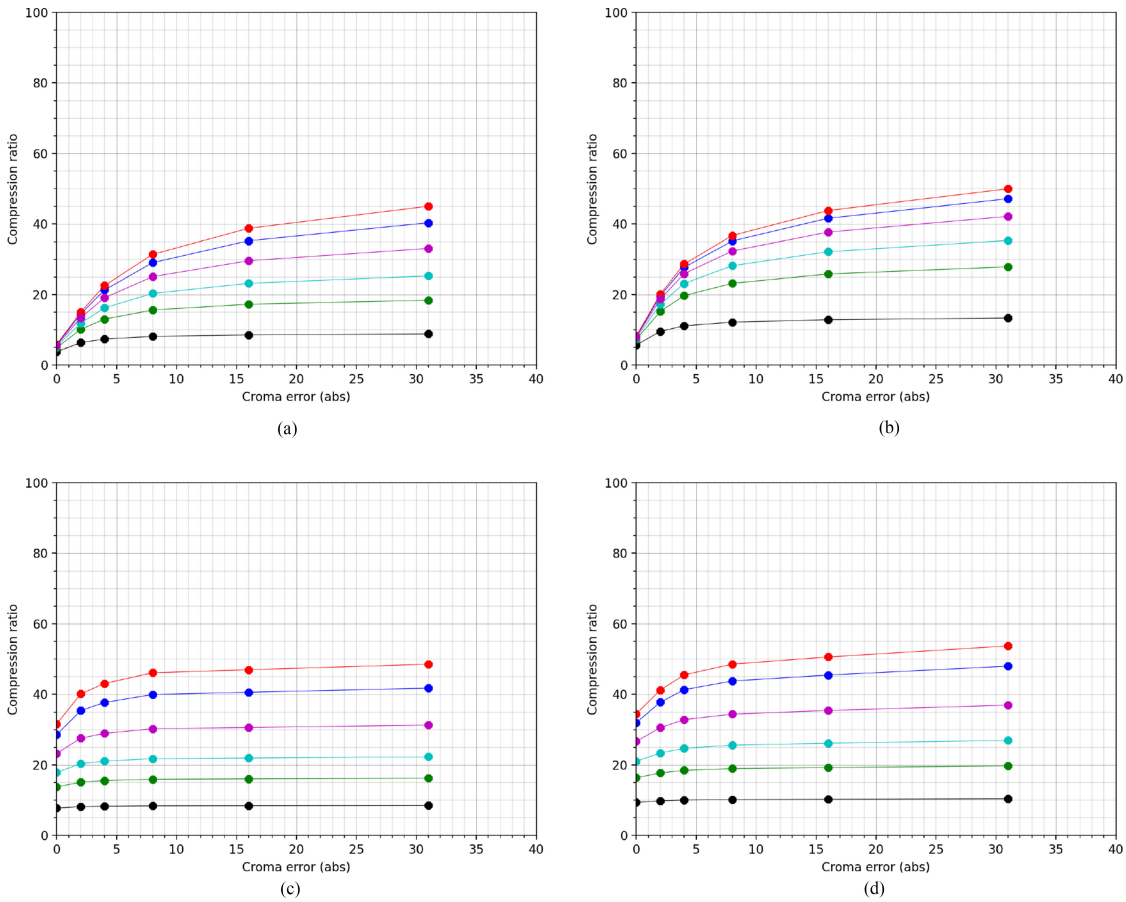


Fig. 6. Relationship of the chroma and luma errors with the compression ratio (Alternative 2). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

spectral bands, as it is specified in the CCSDS 123.0-B-2 standard), and independently compressing each color channel as a grayscale video [26].

B. Transformation to the YCbCr Domain

Although the previous strategy is viable for RGB videos, its compression performance is limited. In addition, that solution is unfeasible for multi- and hyperspectral video, since an independent compression instance is required for each spectral channel, incurring in an area consumption overhead when the number of spectral channels increases significantly. Therefore, further optimizations can be added to increase the overall compression performance, also reducing the resource utilization when this approach is implemented on hardware, compared to the approach proposed in Section III-A. In order to keep the compression solution fully compliant with the CCSDS 123.0-B-2 standard and to make it flexible so that it can be adapted to multiple situations and requirements, each optimization added here on is implemented into the entire compression chain as an individual unit (i.e., preprocessing stages), which can be optionally used attending to the necessities of the targeted application, resulting in three different compression alternatives. Fig. 3 graphically describes the entire compression chain, including these three compression alternatives and the individual blocks that encompasses them, which are also detailed next.

1) *RGB to YCbCr spectral transformation:* This block carries out the spectral transformation of each video frame from the RGB to the YCbCr color space. By doing so, while each video frame still has three bands, most of its spatial information is concentrated into the first one, the luma (Y), which represents the brightness of the scene, while the chroma bands (Cb and Cr) stores the color information. The conversion is done by applying (12)–(14), as described in the Recommendation ITU-R BT.709-6 that defines the conversion coefficient values for ultrahigh-definition television systems [27]. This kind of transformation is commonly used in most of the standard video compressors, such as the H.264 specification [28], to introduce a higher level of losses in the Cb and Cr components, since they store information less perceptible to the human eye, allowing in this way to increase compression performance without degrading reconstructed video quality. Similarly, a higher absolute and/or relative error can be applied in the CCSDS 123.0-B-2 compressor when using it to individually compress the subsequent Cb and Cr frames, while compressing the Y frames with a lower error or even in lossless mode. This preprocessing stage is graphically described in the second row of Fig. 3, and

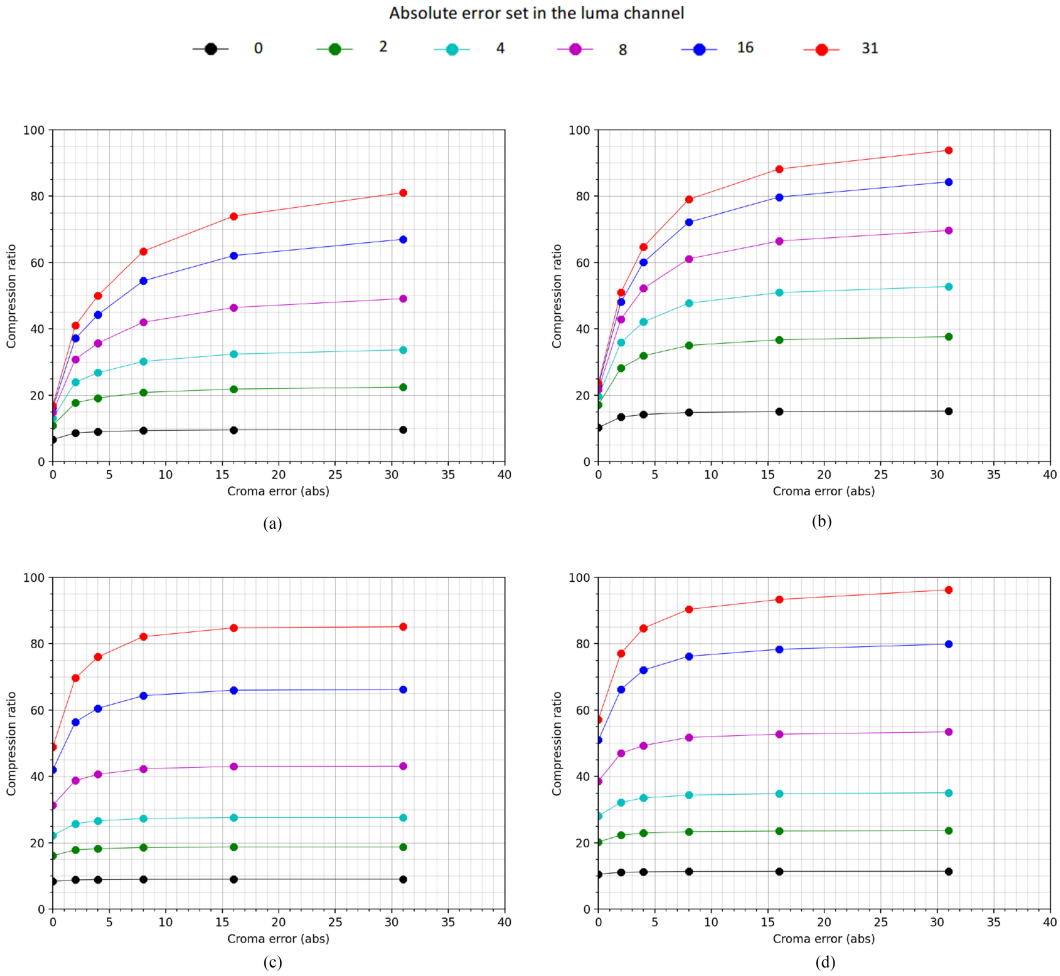


Fig. 7. Relationship of the chroma and luma errors with the compression ratio (Alternative 3). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

it is only executed in the compression alternatives 2 and 3

$$Y = 0.2627 \cdot R + 0.6780 \cdot G + 0.0593 \cdot B \quad (12)$$

$$C_B = \frac{B - Y}{1.8814} \quad (13)$$

$$C_R = \frac{R - Y}{1.4746}. \quad (14)$$

2) *Spatial subsampling of the Cb and Cr bands:* In addition to the previous step, a spatial subsampling of the Cb and Cr channels can be applied by using a bilinear interpolation, reducing in this way their width and height. This strategy is also commonly applied in most of the standard video compressors in order to increase the overall compression ratio achieved at the cost of introducing some spatial blurring in the Cb and Cr channels. Since most of the spatial information is contained in the Y channel, this procedure considerably increases the compression ratio achieved without introducing significant distortions. One further advantage of applying this preprocessing stage is that the computational burden of the subsequent compression stages for the Cb and Cr channels is reduced, decreasing at the same time the resource utilization needed on the final hardware implementation. In this specific work, the size of the Cb and Cr channels has been reduced by a factor of 2 in both their width and height dimensions, thus lessening the overall amount of data to be further processed from these channels by a factor of 4. By introducing this stage in the compression chain, it is also possible to adjust the global latency, being able to compress the luma channel with a processing instance, at the same time that both chroma components are compressed one after the other by using an additional compression core. This preprocessing stage is graphically described in the last row of Fig. 3, and it is only executed in the last compression alternative.

IV. Experimental Results

Multiple experiments have been carried out to validate the compression performance of the developed compression chain at its different configurations. Next, the compression of RGB video sequences, using both the RGB to YCbCr transformation and the spatial subsampling of the Cb and Cr bands, is deeply analyzed, since this configuration is the one that achieves the highest compression ratio. The verification has been automated to accelerate the process, creating a Python-based test framework that

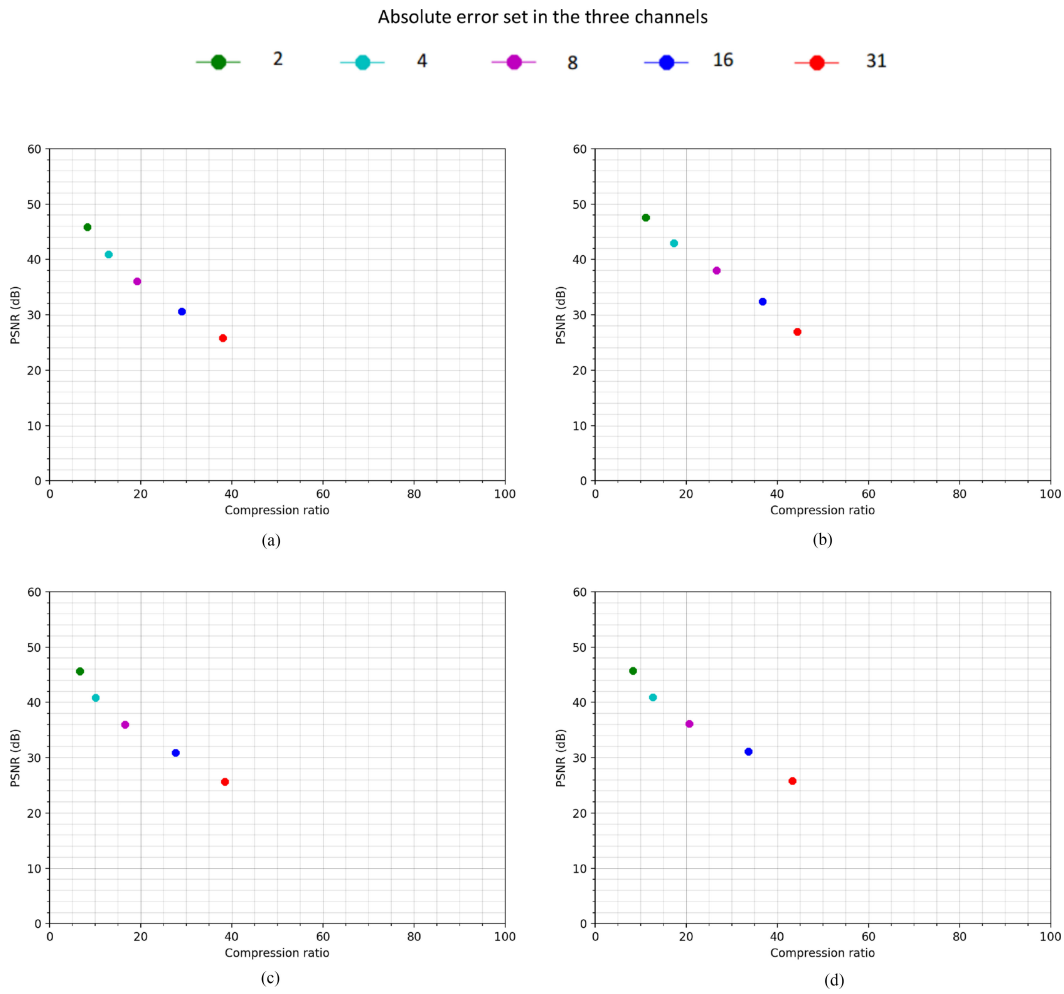


Fig. 8. Relationship between the compression ratio and video quality, measured by PSNR, for the original RGB video (Alternative 1). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

takes a configuration, compressing and decompressing the input video sequence with it, and generating the reconstructed video to analyze it visually. In addition, different reports are generated for each test case, including compression ratio and distortion.

A. Dataset

Four video sequences have been used for carrying out the quality assessment of the CCSDS 123 standard for RGB video compression. The first two sequences have been acquired in our facilities using an IDS uEye sensor [29], providing scenes comprised by 100 frames that are stored using 8 bits per pixel, having a spatial resolution of 1280×1024 pixels. Fig. 4 shows a single frame of each of the captured scenes. They were captured trying to simulate a remote sensing application, where an object with a constant movement is appreciated, while the rest of the scene is almost static, representing an approximate situation to the real scenario. Two additional sequences extracted from the Stanford Drone Dataset have also been employed, denoted as *DeathCircle* and *Nexus* sequences. These videos have also been trimmed to 100 frames, with a spatial resolution of 1400×1904 pixels and a precision of 8 bits per pixel. Unlike video sequences acquired in our facilities, videos from the Stanford dataset present a high local movement, helping to characterize the behavior of the compression solution under demanding scenes, in terms of displacement in adjacent pixels between consecutive frames. Fig. 5 shows a single frame of the two different scenes of the Stanford Drone dataset used for verifying the proposed video compression solution.

B. Overall Solution Assessment

After an exhaustive parameter tuning to characterize the different CCSDS 123.0-B-2 compression parameters, the configuration summarized in Table I is used, since it is the one that provides the best results in terms of compression rate-distortion ratios not only for RGB videos, but also for grayscale video compression, as reflected in [26]. Full prediction mode has been selected, since it was identified as the best option not only for eminently static scenes, but also for the ones with a considerable local movement. This is because it considers a pixel vicinity for the prediction in the current frame, in addition to the pixel at the same position than the current one but in P previous frames, which is the only one employed under reduced mode. Regarding

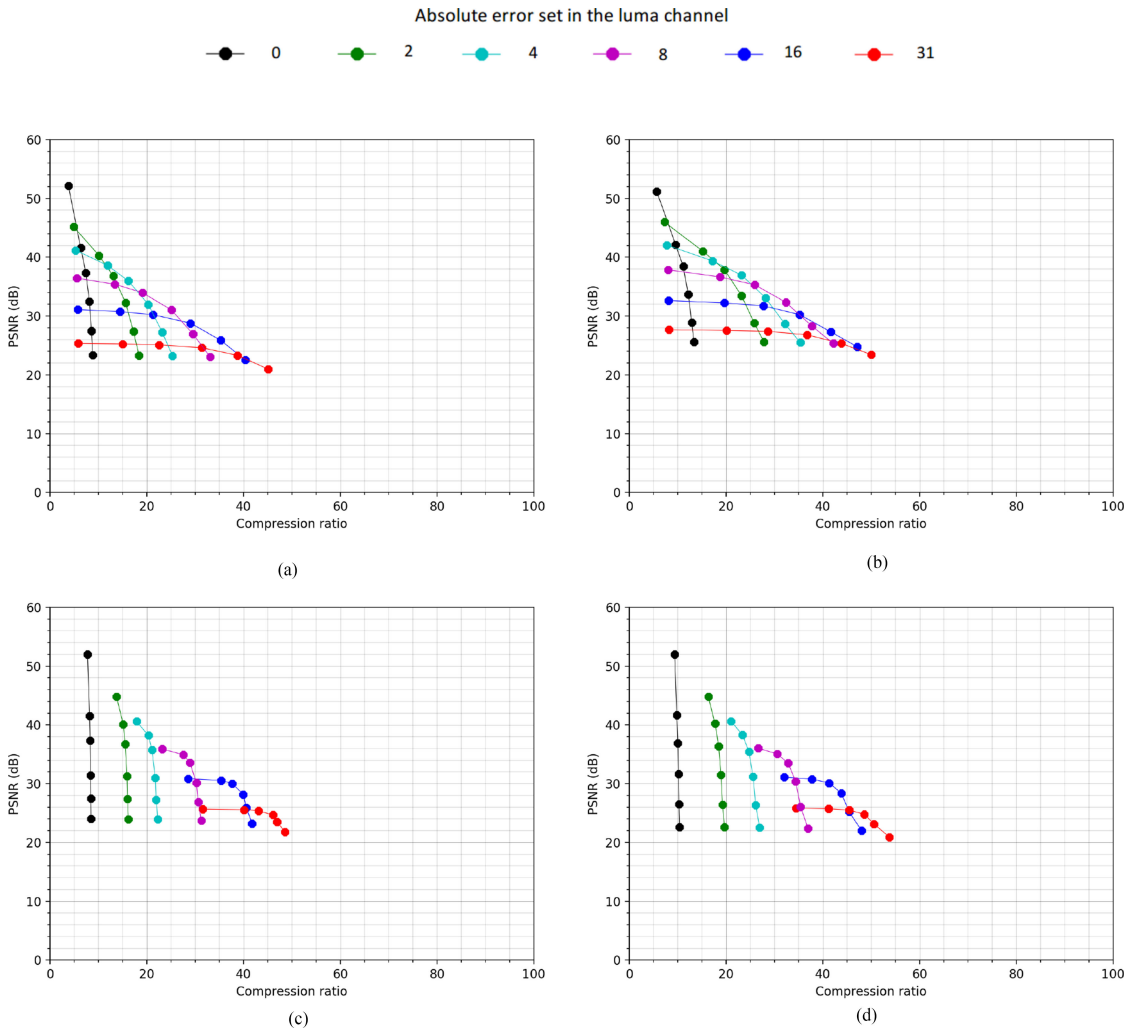


Fig. 9. Relationship between the compression ratio and video quality, measured by PSNR, after applying RGB to YCbCr transformation (Alternative 2). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

the local sum method, the wide neighbor-oriented local sum outperforms the other available options, although the penalty by using of the narrow neighbor-oriented one is around 10–15%. This latter option may be preferable to prioritize the throughput instead of the compression ratio in hardware implementations, since it reduces data dependencies by not taking into account the sample at the left in the current frame. Results in [26] also demonstrate that P values higher than 3 do not provide a compression improvement. At the same time, employing a low P value simplifies hardware implementation, since some memory elements are dependent on this parameter.

Entering in the performance assessment, a sweep is first performed to analyze the impact in the compression ratio of the applied absolute errors on the luma and chroma channels, using values in the range $0 \leq A \leq 31$ in steps of power of 2. This study has been performed for Alternatives 2 and 3, since both of them apply RGB to YCbCr spectral transformation, and obtained results are shown in Figs. 6 and 7, respectively. The case $A = 0$ represents lossless compression, achieving compression ratios around 5 for IDS videos, and between 8 and 10 for Stanford sequences under Alternative 2. Compression ratio in the lossless mode is improved under Alternative 3, reaching up to 10 for IDS videos, and between 9 and 12 for the Stanford sequences.

As shown in Fig. 6, chroma errors slightly increase the compression ratio for all the images in the dataset, while in Fig. 7 is appreciated that compression performance stalls when high error values are applied to chroma components, especially when low absolute error values are applied to the luma. This is due to the fact that the amount of Cr and Cb data to be compressed is just the half than the Y data due to the spatial subsampling. Additionally, Cb and Cr channels present lower entropy and can be more efficiently compressed even with low maximum errors, resulting in less compressed information and becoming negligible in comparison to the part of the compressed bitstream corresponding to the Y channel. On the other side, the luma error emerges as the key value to increase the compression performance, allowing maximum compression ratios for the IDS sequences between 37 and 94 when $A_Y = 2$ and $A_Y = 31$, respectively, under Alternative 3. Similar results are obtained for the Stanford sequences, reaching maximum compression ratios between 24 and 96 when $A_Y = 2$ and $A_Y = 31$, respectively. These results are obtained for the four video sequences by fixing

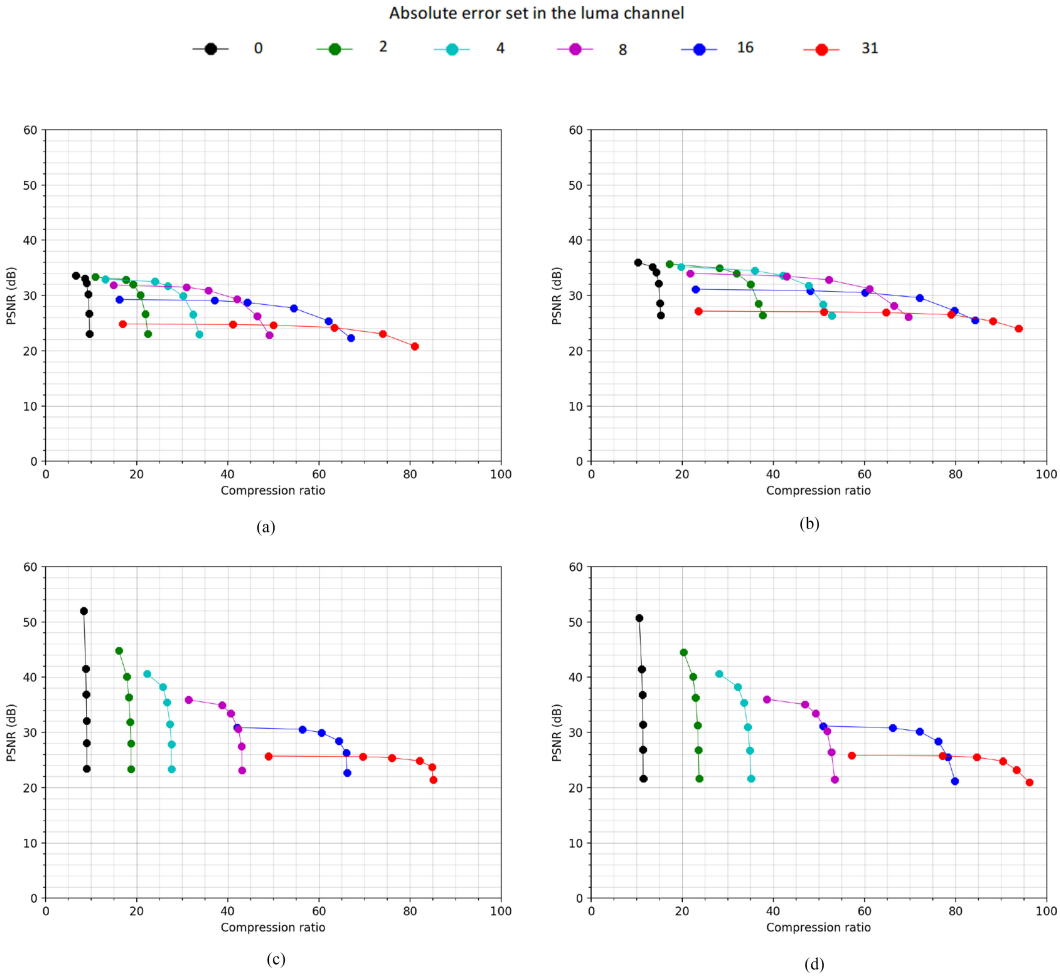


Fig. 10. Relationship between the compression ratio and the video quality, measured by PSNR, after applying RGB to YCbCr transformation and chroma subsampling (Alternative 3). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

both chroma errors to the maximum allowed value of 31. These results are considerably worse under Alternative 2, obtaining maximum compression ratios for the IDS sequences between 27 and 50 when $A_Y = 2$ and $A_Y = 31$, respectively, and between 20 and 54 for the Stanford videos when $A_Y = 2$ and $A_Y = 31$, respectively.

Results in terms of compression ratio versus reconstructed video quality, measured in terms of PSNR, are shown in Figs. 8–10 for Alternatives 1–3, respectively. In these tests, only frame-independent absolute errors are used, since the same tendency is observed when relative errors or both options are applied at the same time. Frame-dependent errors can be used in case that some frames should be preserved with higher level of detail than others (e.g., for target detection and tracking applications). In the case of Alternative 1, the same absolute error is applied to each color channel, since in this approach, the RGB to YCbCr spectral transformation is not performed, compressing directly RGB video in raw format. Purely lossless situation ($A_{Y,Cb,Cr} = 0$) is not shown for Alternative 1, since PSNR tends to infinite. As can be seen for all the experiments performed, the decompressed video quality decreases as the maximum error fixed for the luma and chroma channels increases. This tendency is more notable in the case of the Stanford sequences when applying low absolute errors to the luma channel ($A_Y \leq 4$), under Alternatives 2 and 3.

In the lossless scenario ($A_{Y,Cb,Cr} = 0$), a considerable video quality is reached for all the analyzed video sequences. Under Alternative 2, around 52 dB is obtained for all the video sequences in the dataset. In the case of Alternative 3, PSNR values are lower, being around 34 and 36 dB for the *Guagua* and *Top* sequences, respectively, and 52 dB for the two videos of the Stanford dataset. In this scenario, the differences between the original video and the decompressed one are only due to the RGB to YCbCr transform (in the case of Alternative 2) and the spatial subsampling carried out in the chroma channels (added in Alternative 3), steps that are also carried out in many other video compressors such as the well-known H.264 specification.

Under Alternative 1, best results in terms of PSNR are obtained, at the expense of a penalty in terms of compression ratio. The maximum compression ratio of 38 is obtained for *Guagua* and *DeathCircle* sequences, while this value is increased up to 44 for *Top* and *Nexus* videos. These results are obtained for $A_{Y,Cb,Cr} = 31$ and guaranteeing a proper visual quality, since PSNR ≥ 26 dB in all the cases.

Compression ratios up to 20 (i.e., 0.4 bits per pixel) and 28 (i.e., approximately 0.29 bits per pixel) are reached for the

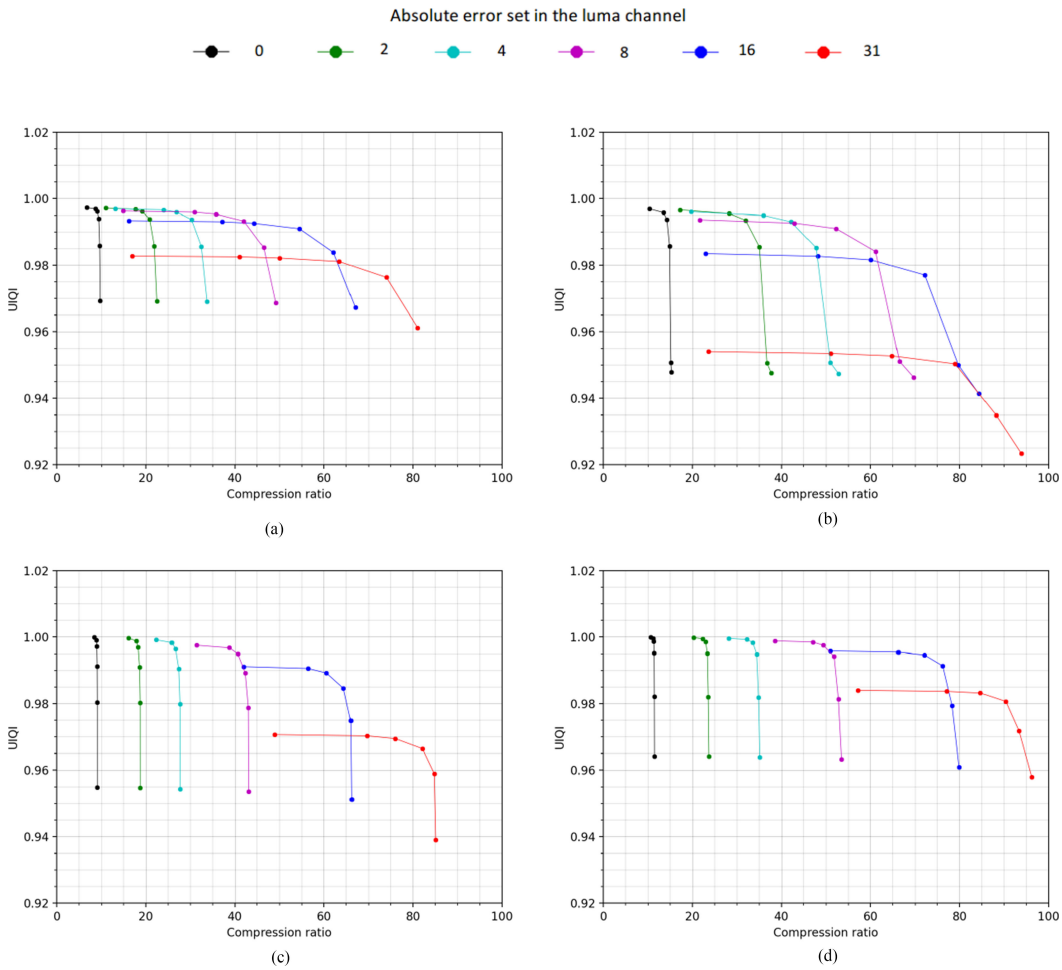


Fig. 11. Relationship between the compression ratio and the video quality, measured by UIQI, after applying RGB to YCbCr transformation and chroma subsampling (Alternative 3). (a) *Guagua* sequence. (b) *Top* sequence. (c) *DeathCircle* sequence. (d) *Nexus* sequence.

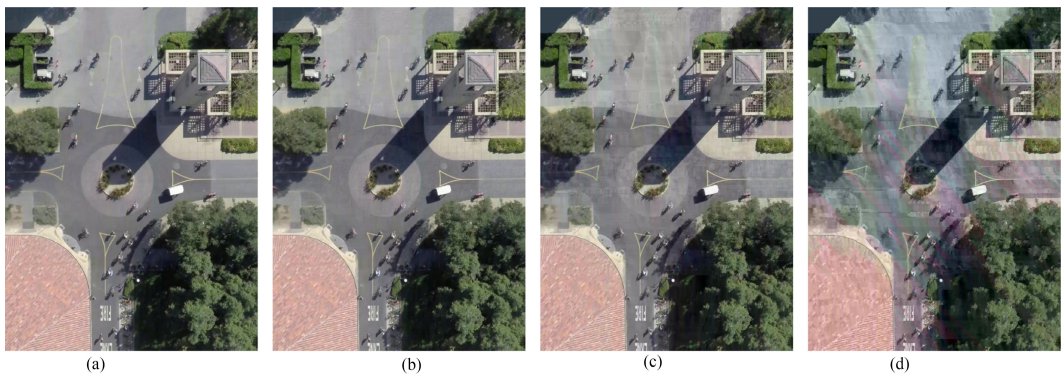


Fig. 12. Visual aspect of the decompressed frames from the *DeathCircle* video sequence when compressing it with different maximum errors values for the Y channel (A_Y) and for the Cb and Cr channels (A_C). (a) $A_Y = 4$ and $A_C = 2$. (b) $A_Y = 8$ and $A_C = 2$. (c) $A_Y = 16$ and $A_C = 4$. (d) $A_Y = 31$ and $A_C = 8$.

Guagua and *Top* sequences, respectively, when fixing $A_Y = 4$ and $A_{Cb,Cr} = 8$ under Alternative 2. For the case of *DeathCircle* and *Nexus* videos, maximum compression ratios of 22 (i.e., 0.36 bits per pixel) and 26 (i.e., around 0.31 bits per pixel) are obtained, respectively, using the same approach and identical error values. In general terms, higher compression ratios are obtained under Alternative 3. Fixing also the error limits to $A_Y = 4$ and $A_{Cb,Cr} = 8$, maximum compression ratios up to 30 (i.e., around 0.27 bits per pixel) and 47 (i.e., 0.17 bits per pixel) are achieved for the *Guagua* and *Top* sequences, respectively. Compression results are around 27 (i.e., around 0.3 bits per pixel) and 34 (i.e., approximately 0.24 bits per pixel) for *DeathCircle* and *Nexus* videos, respectively, by applying the aforementioned errors to the three spectral channels. Under this configuration,

Fig. 13. Visual aspect of the decompressed frames from the *Guagua* video sequence when compressing it with different maximum errors values for the Y channel (A_Y) and for the Cb and Cr channels (A_C). (a) $A_Y = 4$ and $A_C = 2$. (b) $A_Y = 8$ and $A_C = 2$. (c) $A_Y = 16$ and $A_C = 4$. (d) $A_Y = 31$ and $A_C = 8$.

compression has not effect over the video quality (approximately 30 dB for IDS sequences and 32 dB for Stanford videos under Alternative 3) from a visually inspection point.

When error limits are increased to $A_Y = 8$ and $A_{Cb,Cr} = 16$, higher compression ratios are achieved. For Alternative 2, maximum compression ratios of 29 and 38 are obtained for *Guagua* and *Top* sequences, respectively. In the case of *DeathCircle* and *Nexus* videos, compression ratios up to 31 and 36 are reached, respectively, specifying same error limits. In all these test cases, video quality is over 25 dB, ensuring visual video fidelity. Compression results under this error configuration but for Alternative 3 achieve maximum values of 47 and 66 for the *Guagua* and *Top* sequences, respectively, while video quality is still over 25 dB in both cases, maintaining visual video fidelity. The same video quality level is achieved by the Stanford sequences applying the same error limits, with compression ratios up to 43 and 53 for *DeathCircle* and *Nexus* sequences, respectively.

In addition to the PSNR, which is a mathematically defined measure of the video quality, we provide also results in terms of UIQI. This metric tries to provide a quality measurement approach independent of the images under test, the viewing conditions, or the observer point of view [30]. Results comparing the compression ratio versus the UIQI are provided in Fig. 11 for Alternative 3. As can be observed, UIQI values higher than 0.92 have been obtained for all the tests performed for the different video sequences in the dataset. These results are considered satisfactory, since these values are close to the maximum allowed (i.e., 1 is just achieved under lossless compression).

From a visual point of view, video degradation is clearly appreciated in Alternative 3 for higher compression ratios (e.g., for $A_Y \geq 16$), although object shapes are still preserved. Some examples are provided in Figs. 12 and 13, where a frame of the *DeathCircle* and *Guagua* sequences is shown, respectively, applying in each case different error levels to the luma and the chroma channels. As reflected in both figures, degradation is not appreciated from a visual point until the third frame, when a slight degradation is observed at the top of the frames applying error values of $A_Y = 16$ and $A_{Cb,Cr} = 4$. Under this configuration, compression ratios of 61 (i.e., 0.13 bits per pixel) and 44 (i.e., 0.18 bits per pixel) are obtained for each video sequence, respectively. PSNR values are around 30 and 28 dB for the *DeathCircle* and *Guagua* sequences, respectively, while UIQI ≥ 0.99 in both cases.

The highest level of degradation is observed in the last frame, when $A_Y = 31$ and $A_{Cb,Cr} = 8$. In this point, brightness losses are combined with color saturation, obtaining a distorted sequence where objects are still clearly identified in the scene. In addition, some artifacts are observed, related to the way in which the prediction is carried out by the CCSDS 123.0-B-2 algorithm. In the full mode, which is the selected one for the tests performed, the predictor uses the previous vertical, horizontal, and diagonal neighbor samples to predict the current one, and

this may result in diagonal artifacts. The level of degradation introduced by these artifacts depends on the absolute error set in the prediction. Since higher errors are being set for the chroma channels in the experiments performed in this article, these artifacts tend to have a higher impact on the image color than on the image shapes, which are preserved by the luma channel. Under this configuration, compression ratios of 83 (i.e., approximately 0.1 bits per pixel) and 64 (i.e., around 0.13 bits per pixel) are obtained for the *DeathCircle* and *Guagua* sequences, respectively, with a PSNR around 25 dB in both cases. Applying this error configuration, UIQI values of 0.967 and 0.981 are obtained for *DeathCircle* and *Guagua* sequences, respectively. After analyzing all these results, the necessity of reaching a tradeoff between the compression ratio and the video quality is remarkable, prioritizing one metric or the other taking into account the target application requirements.

V. Conclusion

In this article, the CCSDS 123.0-B-2 algorithm for near-lossless compression of multi- and hyperspectral images has been adapted to compress RGB video sequences while being still fully compliant with the standard and without modifying its core functionality. The goal of this approach is to provide a solution for remote sensing applications that allows us to carry out the compression of data of different nature with a single compression core that can be efficiently executed on-board satellites. To do this, the followed approach consists of using the temporal domain to predict the information of the subsequent video frames instead of the previous spectral channels, as it is done in the CCSDS 123.0-B-2 standard. In order to achieve this goal without modifying the compressor, the RGB video to be compressed has to be split into three data sequences, one per color channel, and independently processed within the CCSDS 123.0-B-2 compressor. Additionally, in order to increase the overall compression performance without modifying the CCSDS 123.0-B-2 compressor, two extra preprocessing stages have been added, including a YCbCr spectral transformation and a spatial subsampling of the chroma channels. These two additional steps allow us to increase the overall compression ratio without significantly decreasing the video quality by taking advantages of the redundancies in the RGB color space. Additionally, these steps also reduce the overall computational burden since the amount of data to be processed by the CCSDS 123.0-B-2 compressor are reduced. Similar spectral transformation-based strategies may be used in future works to allow an efficient usage of this approach for compressing multi- and hyperspectral video sequences.

The proposed solution has been validated using four different video sequences representative from a real remote sensing scenario, where the overall scene preserves relatively stable but local movement is present due to moving objects. Many tests have been carried out providing a deep analysis of the compression performance at different compression ratios, also evaluating the quality of the decompressed video sequences both quantitatively and qualitatively after each preprocessing stage. The obtained results demonstrate the goodness of the proposed solution for remote sensing on-board applications, having achieved compression ratios up to 39 (i.e., 0.21 bits per pixel) in the experiments carried out in this article without observing any degradation by visual inspection (i.e., almost lossless). Higher compression ratios can be achieved at the cost of decreasing the decompressed video quality. While, in this case, the present objects are still clearly distinguishable in the decompressed video, the tradeoff between the compression ratio and the decompressed video quality has to be set to meet the targeted application requirements.

Further future works may also include the modification of the proposed CCSDS 123.0-B-2 predictor for being able to work with regions of interest (ROIs), defining a second error parameter applied only in this ROI. This latter strategy allows us to specify at a pixel level the relevant spatial areas that need to be preserved with higher level of detail and the areas that can be more aggressively compressed. In addition, different approaches are currently under study to extend the proposed solution for multispectral video compression. Among the alternatives under analysis, we are considering both alternative transform-based approaches and stages based on lightweight convolutional networks, taking into account on-board hardware restrictions.

References

[1] Y. Luo, L. Zhou, S. Wang, and Z. Wang, "Video satellite imagery super resolution via convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2398–2402, Dec. 2017.

[2] X. Cao, X. Tong, Q. Dai, and S. Lin, "High resolution multispectral video capture with a hybrid camera system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 297–304.

[3] C. Yang *et al.*, "An airborne multispectral imaging system based on two consumer-grade cameras for agricultural remote sensing," *Remote Sens.*, vol. 6, no. 6, pp. 5257–5278, Jun. 2014.

[4] W. Zhu, Q. Du, and J. E. Fowler, "Multitemporal hyperspectral image compression," *IEEE Geosci. Remote Sens. Lett.*, vol. 8, no. 3, pp. 416–420, May 2011.

[5] H. Shen, Z. Jiang, and W. Pan, "Efficient lossless compression of multitemporal hyperspectral image data," *J. Imag.*, vol. 4, no. 12, Dec. 2018, Art. no. 142.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[7] "Consultative Committee for Space Data Systems, Digital Motion Imagery, CCSDS 766.1-B-3," Blue Book ed. Reston, VA, USA: CCSDS, Apr. 2021.

[8] M. Bayat, L. Rongke, L. Arman, and H. Zarrini, "Earth observation satellite learning to video compressor complexity reduction," in *Proc. IEEE Aerosp. Conf.*, 2020, pp. 1–8.

[9] G. Pastuszak and M. Jakubowski, "Hardware implementation of adaptive motion estimation and compensation for H.264/AVC," in *Proc. Picture Coding Symp.*, 2012, pp. 369–372.

[10] M. Kthiri, P. Kadionik, H. Lévi, H. Loukil, A. B. Atitallah, and N. Masmoudi, "An FPGA implementation of motion estimation algorithm for H.264/AVC," in *Proc. 5th Int. Symp. I/V Commun. Mobile Netw.*, 2010, pp. 1–4.

[11] D. Gonzalez, G. Botella, S. Mokheerje, and U. Meyer-Base, "FPGA-based acceleration of block matching motion estimation techniques," in *Proc. 21st Int. Conf. Field Programmable Log. Appl.*, 2011, pp. 389–392.

[12] K. Laidi and M. Nibouche, "On the performance of FPGA implementation of block matching algorithms for video motion estimation," in *Proc. Int. Conf. Elect. Sci. Technol. Maghreb*, 2018, pp. 1–5.

[13] X.-Y. Li and F. Ji, "A parallel H.264 intra-frame prediction decision architecture based on FPGA," in *Proc. Int. Conf. Comput. Inf. Sci.*, 2013, pp. 1611–1615.

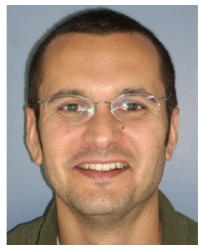
[14] Z. Li, J. Li, Y. Zhao, C. Rong, and J. Ma, "A SoC design and implementation of H.264 video encoding system based on FPGA," in *Proc. 6th Int. Conf. Intell. Human- Mach. Syst. Cybern.*, 2014, vol. 2, pp. 321–324.
[15] G. Pastuszak, "FPGA architectures of the quantization and the dequantization for video encoders," in *Proc. 17th Int. Symp. Des. Diagnostics Electron. Circuits Syst.*, 2014, pp. 290–293.
[16] G. Pastuszak, "A high-performance architecture of the double-mode binary coder for H.264.AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 7, pp. 949–960, Jul. 2008.
[17] V. Rosa, L. Max, and S. Bampi, "High performance architectures for the arithmetic encoder of the H.264/AVC CABAC entropy coder," in *Proc. 17th IEEE Int. Conf. Electron., Circuits Syst.*, 2010, pp. 383–386.
[18] T. Wang, C.-K. Chen, Q.-H. Yang, and X.-A. Wang, "FPGA implementation and verification system of H.264/AVC encoder for HDTV applications," in *Advances in Computer Science and Information Engineering*, D. Jin and S. Lin, Eds. Berlin, Germany: Springer, 2012, pp. 345–352.
[19] M. Li, W. Zuo, S. Gu, J. You, and D. Zhang, "Learning content-weighted deep image compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3446–3461, Oct. 2021.
[20] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3214–3223.
[21] *Low-Complexity Lossless and Near-Lossless Multispectral and Hyperspectral Image Compression, CCSDS 123.0-B-2*, vol. 2, Blue Book ed., Consultative Committee for Space Data Systems, Reston, VA, USA, Feb. 2019.
[22] M. Hernandez-Cabronero *et al.*, "The CCSDS 123.0-B-2 low-complexity lossless and near-lossless multispectral and hyperspectral image compression standard: A comprehensive review," *IEEE Geosci. Remote Sens. Mag.*, to be published, doi: 10.1109/MGRS.2020.3048443.
[23] I. Blanes, A. Kiely, M. Hernández-Cabronero, and J. Serra-Sagristà, "Performance impact of parameter tuning on the CCSDS-123.0-B-2 low-complexity lossless and near-lossless multispectral and hyperspectral image compression standard," *Remote Sens.*, vol. 11, no. 11, Jun. 2019, Art. no. 1390.
[24] *Lossless Data Compression, CCSDS 1210-B-3*, vol. 1, Blue Book ed., Consultative Committee for Space Data Systems, Reston, VA, USA, Aug. 2020.
[25] *Lossless Multispectral and Hyperspectral Image Compression, Informational Report CCSDS 120.2-G-1*, Green Book ed., Consultative committee for space data systems, Reston, VA, USA, Dec. 2015.
[26] Y. Barrios, R. Guerra, S. López, and R. Sarmiento, "Performance assessment of the CCSDS-123 standard for panchromatic video compression on space missions," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022, Art. no. 5507905, doi: 10.1109/LGRS.2021.3099032.
[27] *Parameter values for ultra-high definition television systems for production and international programme exchange*, vol. 1, Recommendation ITU-R BT.2020-2, 2015.
[28] *Advanced Video Coding for Generic Audiovisual Services*, Recommendation ITU-T H.264, Jun. 2019 (Series H: Audiovisual and Multimedia Systems. Infrastructure of audiovisual services, coding of moving video).
[29] *uEye Cameras, User Manual*, 3rd ed., IDS Imaging Development Systems GmbH, Obersulm, Germany, Mar. 2009.
[30] Z. Wang and A. Bovik, "A universal image quality index," *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, Mar. 2002.

Yubal Barrios was born in Las Palmas de Gran Canaria, Spain, in 1993. He received the bachelor's and M.S. degrees in telecommunications engineering in 2016 and 2017, respectively, from the University of Las Palmas de Gran Canaria, Las Palmas de Gran Canaria, where he is currently working toward the Ph.D. degree with the Institute for Applied Microelectronics (IUMA).

In 2017, he was funded by the IUMA, where he has conducted his research activities at the Integrated Systems Design Division in the context of hardware implementations for hyperspectral image compression on field-programmable gate arrays and multiprocessor systems on a chip, applying both high-level synthesis and register-transfer level design methodologies. In 2019, he was invited as a Visiting Researcher by the Microelectronics Section of the European Space Research and Technology Centre, core of the European Space Agency, Noordwijk, The Netherlands. His current research interests include the development of efficient algorithms for on-board hyperspectral image compression and reconfigurable hardware architectures optimized in terms of throughput, memory usage, and power consumption.

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

He is currently with the Institute for Applied Microelectronics, University of Las Palmas de Gran Canaria. He was funded by the Institute of Applied Microelectronics to do his Ph.D. research with the Integrated System Design Division. In 2016, he was a Researcher with the Configurable Computing Lab, 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 (Senior Member, IEEE) 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, 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 articles 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 Active Reviewer for different JCR journals and a Program Committee Member of a variety of reputed international conferences. He was a Program Chair of the IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing in 2014 and the SPIE Conference of High Performance Computing in Remote Sensing from 2015 to 2018. He is an Associate Editor for the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, MDPI Remote Sensing, and Mathematical Problems in Engineering. He was an Associate Editor for the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS from 2008 to 2013. He is also a Guest Editor for different special issues in JCR journals related to his research interests.

Roberto Sarmiento is currently a Full Professor of Electronic Engineering with the School of Electronics and Telecommunication Engineering, University of Las Palmas de Gran Canaria (ULPGC), Las Palmas de Gran Canaria, Spain. He contributed to setting up the School of Electronics and Telecommunication Engineering, ULPGC, where he was the Dean of the Faculty from 1994 to 1998 and the Vice-Chancellor for Academic Affairs and Staff from 1998 to 2003. He is a co-founder of the Research Institute for Applied Microelectronics, ULPGC, and the Director of the Integrated Systems Design Division of this institute. He has authored or coauthored more than 90 journal papers and more than 160 conference papers. He has participated in more than 60 projects and research programs funded by public and private organizations. His current research interests include electronic system on-board satellites.

Prof. Sarmiento has been awarded with five to six years research periods by the National Agency for the Research Activity Evaluation in Spain.