

IMAGE AND RADIO-FREQUENCY DATA COMPRESSION FOR OPS-SAT USING FAPEC

Jordi Portell^(1,2,3,4), Aniol Martí^(1,5), Riccardo Iudica⁽¹⁾, David Evans⁽⁶⁾, Vladimir Zelenevskiy^(6,7)

⁽¹⁾*DAPCOM Data Services S.L.*

c. dels Vilabella, 5-7. 08500 Vic (Barcelona), Spain. Email: jordi.portell@dapcom.es

⁽²⁾*Institut de Ciències del Cosmos (ICCUB), Universitat de Barcelona (UB)*

c. Martí i Franquès, 1. 08028 Barcelona, Spain

⁽³⁾*Departament de Física Quàntica i Astrofísica (FQA), Universitat de Barcelona (UB)*

c. Martí i Franquès, 1. 08028 Barcelona, Spain

⁽⁴⁾*Institut d'Estudis Espacials de Catalunya (IEEC)*

c. Gran Capità, 2-4. 08034 Barcelona, Spain

⁽⁵⁾*Department of Signal Theory and Communications, Technical University of Catalonia (SPCOM/UPC)*

Campus Nord, c. Jordi Girona 1-3, 08034 Barcelona, Spain

⁽⁶⁾*European Space Operations Center (ESOC), European Space Agency (ESA)*

Robert-Bosch Strasse, 5. 64293 Darmstadt, Germany

⁽⁷⁾*Telespazio Germany GmbH*

Europaplatz, 5. 64293 Darmstadt, Germany

ABSTRACT

OPS-SAT is an ESA technology demonstration cubesat which includes a colour camera, a Software Defined Radio (SDR) receiver and a powerful ARM processor. One of the experiments executed there is FAPEC, a high-performance and versatile data compression software. Among others, it features image compression and linear prediction coding algorithms, suitable for multi-band and baseband radio-frequency (RF) samples, respectively. Since its deployment on-board OPS-SAT in late 2020, FAPEC has allowed for downloading a large set of Earth Observation images. Recently, thanks to ESA Open Space Innovation Platform funds, these two algorithms from FAPEC are being improved to get better compression ratios and speeds, add video compression capabilities, and higher quality levels in case of lossy compression. A smart lossy approach is being developed for radio-frequency data, identifying the time segments with signal presence and quantizing noise-only samples to further reduce the size. In order to identify the really useful files to be downloaded from the satellite, on-board data analysis capabilities are being developed as well. In this work we present in-orbit results, recent developments and preliminary results obtained with the new algorithms on real data.

1. INTRODUCTION

During the last decade, we have witnessed the raise of the *NewSpace* paradigm. It is a new philosophy of space missions, which are often based on the cubesat standard [1] and a *constellation* of several satellites, aiming at a higher overall reliability and reduced revisit time. The use of cubesats, thanks to their much reduced cost with respect to medium or large satellites, also allows for educational, experimental or technology demonstration missions. One of these is OPS-SAT [2-4], a technology demonstration 3U cubesat of the European Space Agency (ESA). It features a high-resolution camera (2048×1944 pixels) with a Bayer Colour Filter Array (CFA) and 12-bit pixels, with a capability to acquire bursts of up to five frames per second [5]. It also includes a Software Defined Radio (SDR) front-end and a patch antenna, able to acquire radio-frequency (RF) signals in the Ultra-High Frequency (UHF) band, delivering 12-bit in-phase and quadrature samples (I/Q). OPS-SAT also has a powerful dual-core ARM microprocessor at a clock speed of 800 MHz and with 1 GB of RAM, which is the core of the Satellite Experimental Processing Platform (SEPP). One of the novelties of this mission is the Nanosat Mission operations Framework (NMF) [6], based on the Java programming language, which can run several applications from the experimenters approved by ESA for in-orbit tests. OPS-SAT has other interesting features, such as a Global Positioning System (GPS) receiver, an Attitude Determination and Control System (ADCS) and an optical receiver. In this work, we will focus on the camera, the SDR, the SEPP and the NMF.

Amongst the 226 registered experiments for OPS-SAT, one of them is FAPEC (Fully Adaptive Prediction Error Coder) [7], a data compression software provided by DAPCOM Data Services. It features a variety of pre-processing or decorrelation algorithms, followed by a high-performance adaptive entropy coding core, altogether supported by a multi-threaded framework providing several command-line interface (CLI) and application process interface (API) options to the user. FAPEC can be invoked from the Linux, macOS or Windows command line, or from a C, Python or Java API.

The latter has made possible its integration in the OPS-SAT NMF, as we will describe hereafter, whereas the CLI has allowed for a prompt and easy integration in the OPS-SAT SEPP in orbit since late 2020.

2. CURRENT FAPEC USAGE IN OPS-SAT

FAPEC is being used by the OPS-SAT operations team in its CLI form, invoking it from the Linux command-line (or shell scripts) to compress the raw image files acquired on-board before downloading them to the ground station. The CILLIC image compression algorithm [8] is used for this purpose, which provides ratio-vs-PSNR figures similar to a Discrete Wavelet Transform (DWT) [9] but with a much faster operation. FAPEC and CILLIC were updated to support the peculiar image format of the OPS-SAT camera, which encodes the 12-bit pixels in the *most* significant bits of 16-bit integer values, contrary to the usual convention. The Bayer CFA format is also supported, which is handled as a multi-band image with 4 bands. Experimental results soon revealed that our spectral decorrelators do not provide any significant improvement, but nonetheless, CILLIC internally tests several inter-band decorrelators (including a spatial-only option) and automatically selects the best one. Finally, CILLIC supports lossless, near-lossless (fixed-quality) and lossy (fixed-rate) operation. The latter is the option typically used on-board, with a target ratio of 10. Fig. 1 shows an execution of the FAPEC CLI on-board OPS-SAT. As can be seen, FAPEC compresses the camera images at a speed of about 9 MB/s (in single-thread mode), that is, less than one second per image.

```
[29-11-2020 17:00:26] COMMAND Uplink to SEPP: for f in /home/exp1000/toGround/edge/*.ims_rgb; do
c='/home/exp100/fapec -q -chunk 512K -mt 1 -dtype 16 -cillic 2048 1944 1 x10 12 4 -lev 5 -ow
-o /home/exp100/toGround/'${basename ${f%.*}}.fapec); eval '$c $f >> /home/exp100/f.log'; done
[29-11-2020 17:00:33] DATA: START
[29-11-2020 17:00:33] DATA: STOP
[29-11-2020 17:00:34] COMMAND Uplink to SEPP: cat /home/exp100/f.log; ls -larthR /home/exp100/toGround
[29-11-2020 17:00:41] DATA: START
[29-11-2020 17:00:41] (...)
[29-11-2020 17:00:41] FAPEC Archiver - 20.0.0 Beta r2280 (2020-11-15)
[29-11-2020 17:00:41] (...)
[29-11-2020 17:00:41] Compressing 1 file into /home/exp100/toGround/img_msec_1606638723330_2.fapec...
[29-11-2020 17:00:41] [1/1] /home/exp1000/toGround/edge/img_msec_1606638723330_2.ims_rgb (7.6 MB)...
[29-11-2020 17:00:41] Done: 7.6 MB compressed to 0.8 MB (ratio 9.9437) in 0.8 seconds (9.3 MB/s)
[29-11-2020 17:00:41] /home/exp100/toGround:
[29-11-2020 17:00:41] -rw-r--r-- 1 root root 806.2K Nov 29 17:00 img_msec_1606601765418_2.fapec
[29-11-2020 17:00:41] -rw-r--r-- 1 root root 782.2K Nov 29 17:00 img_msec_1606638723330_2.fapec
```

Fig. 1. Screenshot of the OPS-SAT SEPP invoking FAPEC to compress two image files

Besides this usage as a simple CLI program with the adequate parameters provided by the user, we have also implemented a Java wrapper for the integration of FAPEC in the OPS-SAT NMF. It is composed of a Java Native Interface (JNI) plus an application following the NMF API, ready to be invoked from it. Left panel of Fig. 2 illustrates this integration, whereas the right panel shows one of the steps in its execution from the NMF front-end.

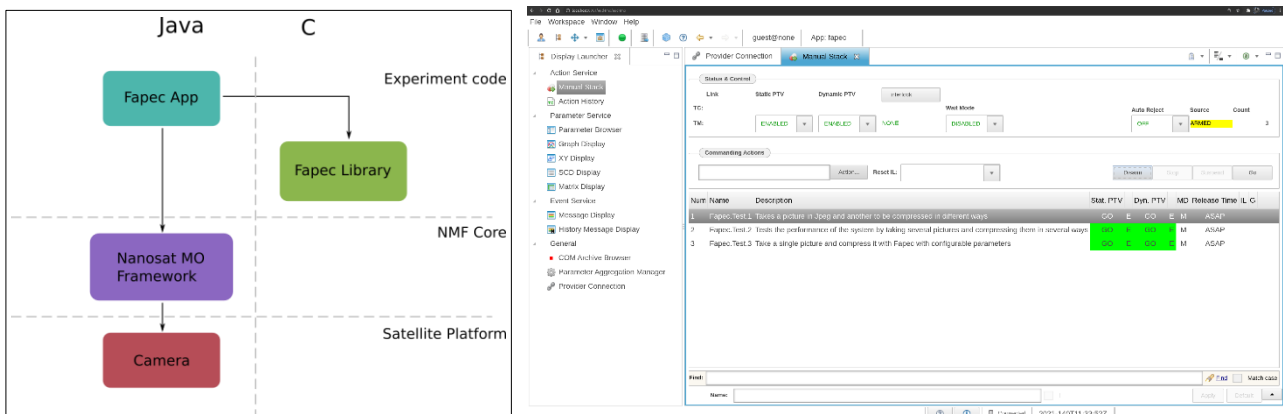


Fig. 2. Integration and execution of FAPEC in the OPS-SAT Nanosat Mission operations Framework (NMF)

3. IMPROVEMENTS IN IMAGE AND RADIO-FREQUENCY COMPRESSION

ESA opened a call in 2021 in the Open Space Innovation Platform (OSIP) asking for ideas to improve OPS-SAT operations. DAPCOM Data Services applied with the proposal “Versatile data compression software for sustained high-throughput in-orbit data acquisition”, which was accepted and is being executed during 2022. In a nutshell, the main goals

were the improvement of the CILLIC and Wave algorithms of FAPEC, aiming at higher throughputs, better compression ratios and, in case of lossy image compression, better quality in the recovered images. In the following subsections we describe the main improvements done in these algorithms.

3.1. Updates in the CILLIC Algorithm

The original CILLIC algorithm [8] is based on blocks of 15×15 pixels and a prediction-based (not transform-based) operation. It is composed of three decorrelators: spatial, spectral and mixed. The spatial decorrelator gives the name to the algorithm (Context Interpolation Lossless and Lossy Image Compressor), as it progressively builds the pixel references which are later used to predict a pixel from the interpolation of its neighbours. The spectral decorrelator essentially predicts a pixel from the previous band plus a common inter-band offset. The mixed decorrelator combines spatial and spectral prediction, estimating each pixel from the previous band plus the average inter-band offsets from the west and north pixels. In its near-lossless mode, residuals are quantized and the pixels are then reconstructed during compression to avoid error propagation.

As part of the OSIP-funded developments we have evolved the CILLIC concept into a second version (CILLIC v2). One of its most remarkable features is the increase of the block size to 17×17 pixels, as illustrated in Fig. 3 for the spatial decorrelator. With this, we can obtain better multi-level predictions, starting with type-1 pixels (shown in black in Fig. 3), some of which are predicted from the north or west neighbour blocks combined with the neighbour type-1 pixels of the same block. For the east and south edges we make use of another novelty available in CILLIC v2, which is the determination of thumbnails, composed of the average values determined for each block (and in each band). Thus, thumbnails have $1/289^{\text{th}}$ the original image resolution, and are always available in the compressed file. Thumbnails are determined ahead of the normal block operations, which allow for using them in the predictions. Thus, for type-1 pixels in the east and south edges, we estimate them from the east and south neighbour thumbnails. These inter-block references provide smoother transitions in lossy mode, thus reducing the block effect at high loss levels. Once the type-1 edges have been handled, the inner type-1 pixels are predicted from the linear interpolation of neighbour pixels. This approach is then continued for type-2 pixels (dark grey in Fig. 3), predicted from interpolations of either type-1 or type-2 pixels. These two pixel types form a grid from which the rest of the pixels (in lighter grey shades) are interpolated. In lossy mode, we apply conservative quantization levels in type-2 and specially type-1 pixels, thus forming a medium-resolution lattice that allows to get reasonably good image details even at high loss levels.

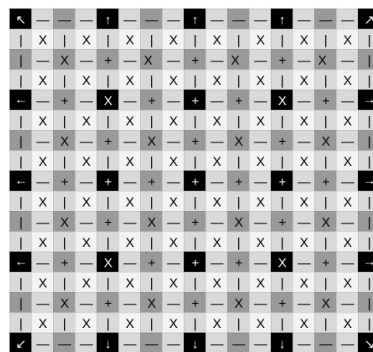


Fig. 3. Schematic of the spatial decorrelation algorithm in CILLIC v2

Inter-band decorrelation is done just with the mixed algorithm for now, that is, predicting each pixel from the same pixel location in the previous band, plus an inter-band offset determined from the north and east pixels of both bands. It provides a fast operation and reasonably good compression performance.

The main novelty of CILLIC v2 is the capability to process raw video files. We have implemented an inter-frame decorrelator based on the same inter-band algorithm plus a motion estimator. The latter is based on a brute-force analysis of the inter-frame Mean Square Error (MSE) at different shifts. That is, we shift the previous-frame block with respect to the current-frame block, within the range of $-17 \dots +17$ pixels (either horizontally or vertically), determine the MSE between adjacent pixels of the whole block, and find the shift leading to the smallest MSE value. We are working on more efficient algorithms, such as hill climbing [10] or phase correlation [11], which would furthermore allow for improvements in the inter-band decorrelation, mainly in case of low-quality cameras with misregistration artefacts.

Given the interest in compressing video files, compression speed is of utmost importance. For this reason we have included the option to use FASEC [12] as the coding stage of CILLIC v2. With this, we can get significantly higher compression throughputs, yet at the cost of a modest reduction in the compression ratios.

Finally, we have revised the data analysis capabilities embedded in the FAPEC framework, which can generate a summary or statistics file with information on each of the *chunks*. In FAPEC, files or buffers are often split into chunks (typically

of uniform size), each 4 KB to 384 MB, which are compressed independently each other. It is an extra reliability layer, as it makes possible to recover healthy chunks from a corrupted file. Also, it brings in the capability to obtain basic statistics or simplistic data analysis on each of the chunks of a file, such as an image. In our case, CILLIC determines the number of blocks in each chunk, the fraction of these handled by each of the decorrelator types (spatial or mixed), and also the fraction of blocks found to be *flat* – that is, with a smaller variability than the quantization step (in case of lossy compression). The latter is remarkably interesting to easily identify images with no relevant content, as well as images with large areas of clouds, sea or space.

3.2. Updates in the Wave Algorithm

The original Wave algorithm of FAPEC [13] is based on Linear Predictive Coding (LPC), that is, a linear filter which predicts samples following the model

$$\hat{x} = \sum_{i=1}^Q h_i x(n-i) \quad (1)$$

where $\hat{x}(n)$ is the predicted sequence, $x(n-i)$ the previous samples, h_i the filter coefficients and Q the filter order. In order to obtain the filter coefficients h_i , we need to solve the well-known Yule-Walker equations. In the Wave implementation, we take advantage of the Levinson-Durbin recursion [14] which reduces the computational complexity of the problem from $O(Q^3)$ to $O(Q^2)$.

There are several reasons that justify using a linear predictor to process RF data. The first one is related to the structure of RF files: 16-bit signed integers in two channels (I/Q). This format is very similar to that of audio files, and LPC is often used for audio compression, for instance in the FLAC algorithm [15]. On the other hand, linear prediction is a very simple method that, following the Ockham's razor, is to be preferred [16]. In addition, a particular case of a RF signals are linearly modulated communications signals. In this case, the Maximum a posteriori (MAP) estimate of the signal can be obtained by using a linear predictor [17], thus the prediction error is very small and the performance of the entropy coder is better. The linear predictor is applied on small independent periods of samples, typically about 8,000 each, but allowing the user to configure up to one million samples. Also, it is able to handle up to 32,000 channels. In addition, a simplistic near-lossless approach is implemented by quantizing the original samples prior to the prediction stage, thus effectively reducing the resolution of the samples.

During the OSIP-funded work, we have verified that this is an optimum compromise approach, offering nearly-optimal coding efficiencies at a reduced computational cost. Remarkably, in lossless mode it is difficult to achieve significantly higher ratios. Nonetheless, we have been able to make some minor improvements, mainly by increasing the maximum filter order to 16, making it adaptive to each period. That is, the Levinson-Durbin recursion stops if the decrease in the error is less than 1% compared with the error from the previous filter order iteration. The default FAPEC block length for this algorithm has also been increased, overall leading to a modest increase in the lossless ratios.

The most significant improvement in the FAPEC Wave algorithm (hereafter “Wave v2”) is the so-called “smart lossy” approach. In cubesats, SDR receivers can often generate data files mostly containing noise. In these cases, we considered that it would be useful to automatically detect the periods that actually contain some sort of signal, keeping a very low level of losses here (or even a lossless mode), and significantly increasing the losses for the noise-only periods. However, signal detection is a challenging problem, especially considering the vast range of use cases of a satellite SDR receiver, including GNSS signals which are based on a spread spectrum approach [18] and are hidden below the noise ground. We have followed and tested two approaches.

First, a very simplistic one based on the prediction error already calculated by the Levinson-Durbin recursion and the accumulated energy from the autocorrelation and the LPC coefficients. In situations where the underlying model is linear, the former can be considered an approximation to the noise level, and similarly, the latter can be taken as the signal level. From these, the quantization level for the samples of that period can be determined, leading to progressively high losses for the periods with a worse signal-to-noise ratio (taken from these estimations).

The second approach is inspired in classical spectrum sensing problems and it involves a higher computational load, although it is probably acceptable given the capacity of the OPS-SAT SEPP. The main idea is to design an energy detector [19] for three different probabilities of false alarm – conservative, moderate and aggressive levels of loss. However, we do not know the noise power, so it has to be estimated [20]. In order to do that, we compute the average periodogram [21] and by invoking the central limit theorem (CLT), the distribution of the periodogram is $N(\sigma_w^2, \sigma_w^4/K)$, where K is the number of segments over which the periodogram is averaged. Assuming that the noise is also Gaussian, we can use the Akaike Information Criterion (AIC) [22] to find which frequency bins are assumed to represent the signal. Thus, we can average the remaining ones to compute an estimate of the noise power.

Similarly to the CILLIC case, these Wave improvements make possible a basic on-the-fly data analysis of SDR data files, allowing to generate short summaries of their contents which, for example, can aid in the decision to download each of the files generated on-board.

4. PRELIMINARY RESULTS

4.1. CILLIC v2 Results

We have implemented most of the improvements designed for CILLIC. The ESA OPS-SAT team has kindly provided us a rich set of test images, some of which were downloaded from the satellite using lossy CILLIC compression and thus with some artefacts that may affect the test results. In Table 1 we show some of these images, with the original image in the left panels and the recovered images in the right panels after applying the maximum near-lossless level in FAPEC. As can be seen, despite of these high loss levels we can still clearly see the overall contents (and usefulness) of the image, even with some detailed features as in row (c). Note that these are still “normal” images (not the thumbnails), simply with highly quantized residuals. It reveals the capability of CILLIC to recover image features even when losing most of the pixels. When providing just the thumbnails, ratios around 500 were observed.

In the central panels of Table 1 we illustrate the basic data analysis capability previously mentioned – namely, the distribution of the flat blocks for each of the images. A very clear example is row (a), where 100% of the blocks are flat for the first chunks (that is, for the upper part of the image), whereas it progressively decreases towards fewer flat blocks for the bottom part. Similarly, row (b) exhibits flat blocks for the central sea areas, and row (d) for the areas with more clouds. Finally, we have also tested the use of FASEC as the entropy coding core, which reveals nearly twice the throughput on an ARM platform.

To test the video compression capability of CILLIC v2 we have used a raw YUV test video from a ground-based scene with a tractor in a field. For now, we only have still images from OPS-SAT taken a few seconds apart, which leads to too high motions for the capabilities of CILLIC. Left panel of Fig. 4 shows one frame of this video, where the camera is following the tractor towards the left. Right panel shows the motion vectors estimated by CILLIC v2 for each of the pixel blocks, showing the vector norm in color. When carefully examining this we can see that the motion is correctly determined, even for the tires of the tractor, where a circular motion can be found. However, for now, when applying this to the inter-frame decorrelator we just get a modest improvement of about 10% in the lossless ratio, probably due to sub-pixel motion levels or due to the simplistic inter-frame decorrelator.

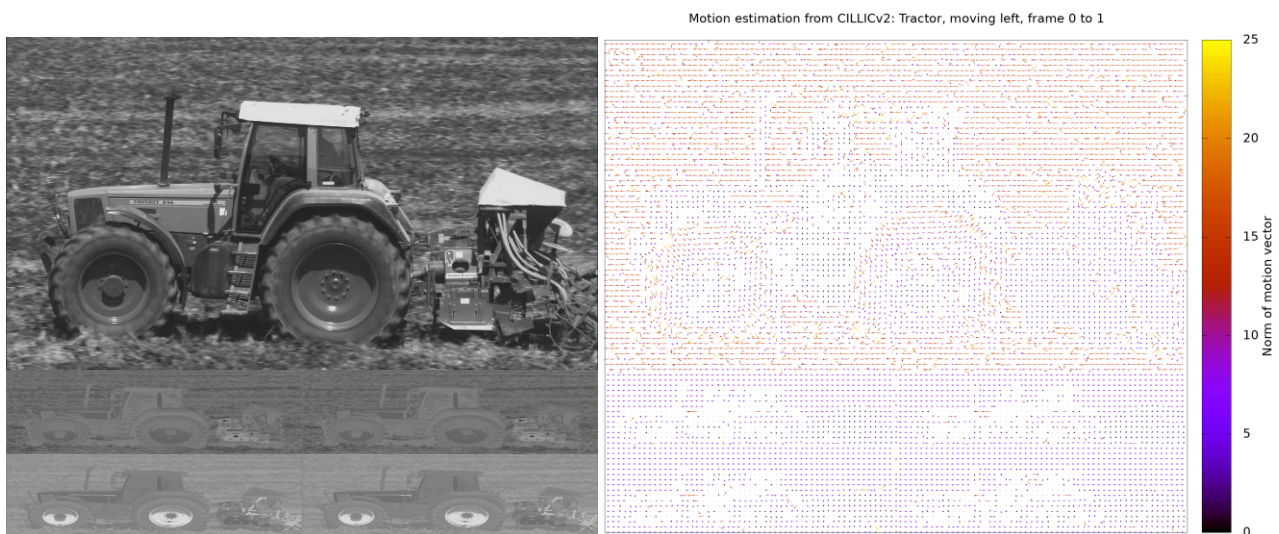



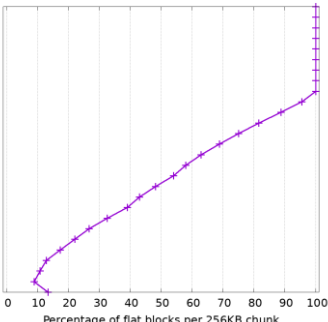

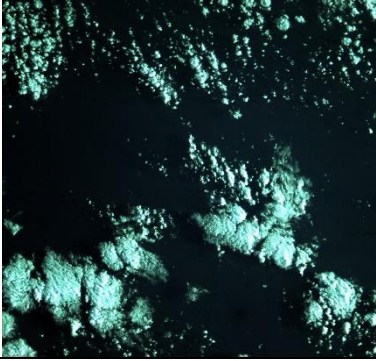
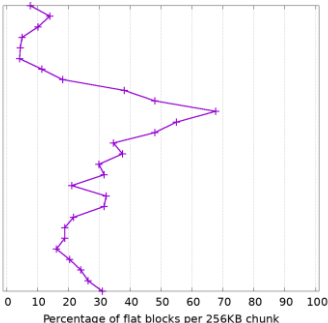
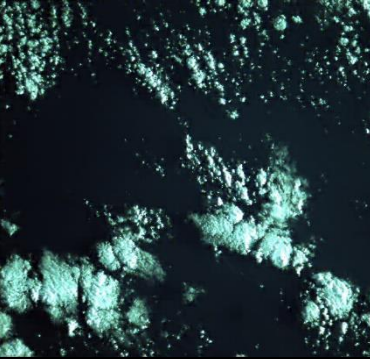
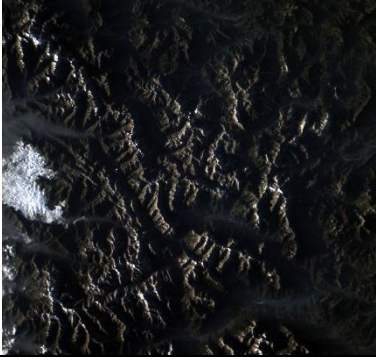
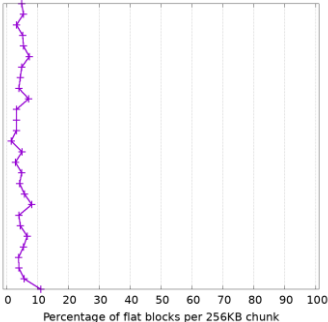
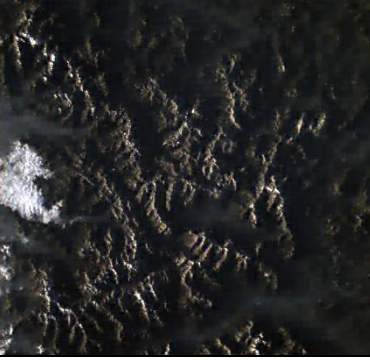
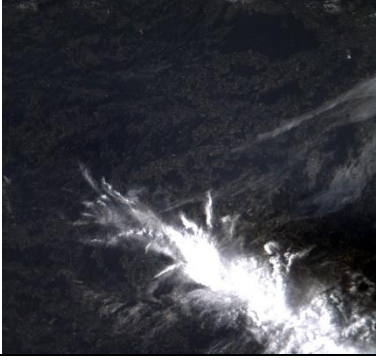
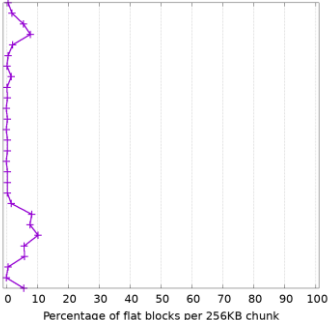

Fig. 4. One frame of the *Tractor* video in raw YUV format (left), and motion estimation vectors from CILLIC (right)

4.2. Wave v2 Results

To test the improvements in Wave v2 we have focused on the new *smart lossy* feature, and more specifically on the signal detection capability. We have taken a real SDR file acquired on-board OPS-SAT during 2022, which reveals quite some signal structure in its spectrogram (top panel of Fig. 5). Remarkably, we can see some signals with a strong Doppler effect, which may presumably correspond to other satellites relatively close to the OPS-SAT orbit.

First, in the central panel of Fig. 5, we show the signal and noise levels estimated by the simplistic approach based on the information from the Levinson-Durbin recursion. By comparing it to the spectrogram, we can see that it follows quite accurately the variation in the signal level, and it even seems to detect the two apparently noisy periods towards the end of the file. Note that those sudden increases and decreases of “noise” may also correspond to GNSS signals.

Table 1. Data compression and analysis tests on OPS-SAT images and FAPEC-CILLIC

Original image	Distribution of flat blocks per chunk (each chunk is 2048×68 pixels, 4 bands)	Recovered image at maximum loss
	<p>(a)</p> <p>Flat blocks per chunk, 1606638723330 (total: 61.2%)</p>  <p>Percentage of flat blocks per 256KB chunk</p>	<p>Ratio 122.8</p> 
	<p>(b)</p> <p>Flat blocks per chunk, 1606689655062 (total: 26.2%)</p>  <p>Percentage of flat blocks per 256KB chunk</p>	<p>Ratio 102.0</p> 
	<p>(c)</p> <p>Flat blocks per chunk, 1654059354810 (total: 5.3%)</p>  <p>Percentage of flat blocks per 256KB chunk</p>	<p>Ratio 45.4</p> 
	<p>(d)</p> <p>Flat blocks per chunk, 1654059402403 (total: 2.6%)</p>  <p>Percentage of flat blocks per 256KB chunk</p>	<p>Ratio 134.0</p> 

Regarding the information-theoretic method based on spectral estimation and AIC (bottom panel of Fig. 5), we can see that the periods highlighted in grey (meaning “signal presence detected”) follow very well the peaks seen in the periodogram, including the “noisy” (or perhaps GNSS) ones mentioned. Other files with similar spectrograms exhibit very similar results. Pending further analysis, these tests seem to reveal that both methods perform quite similarly, with the latter giving more reliability and generalizing better for different kinds of files.

When using the results from the central panel to determine a variable quantization step (keeping at least 4 bits for the noise and 8 bits for the signal), we find very interesting improvements in the ratios. For example, we can move from a ratio of 1.47 (lossless) to 2.37 (smart lossy).

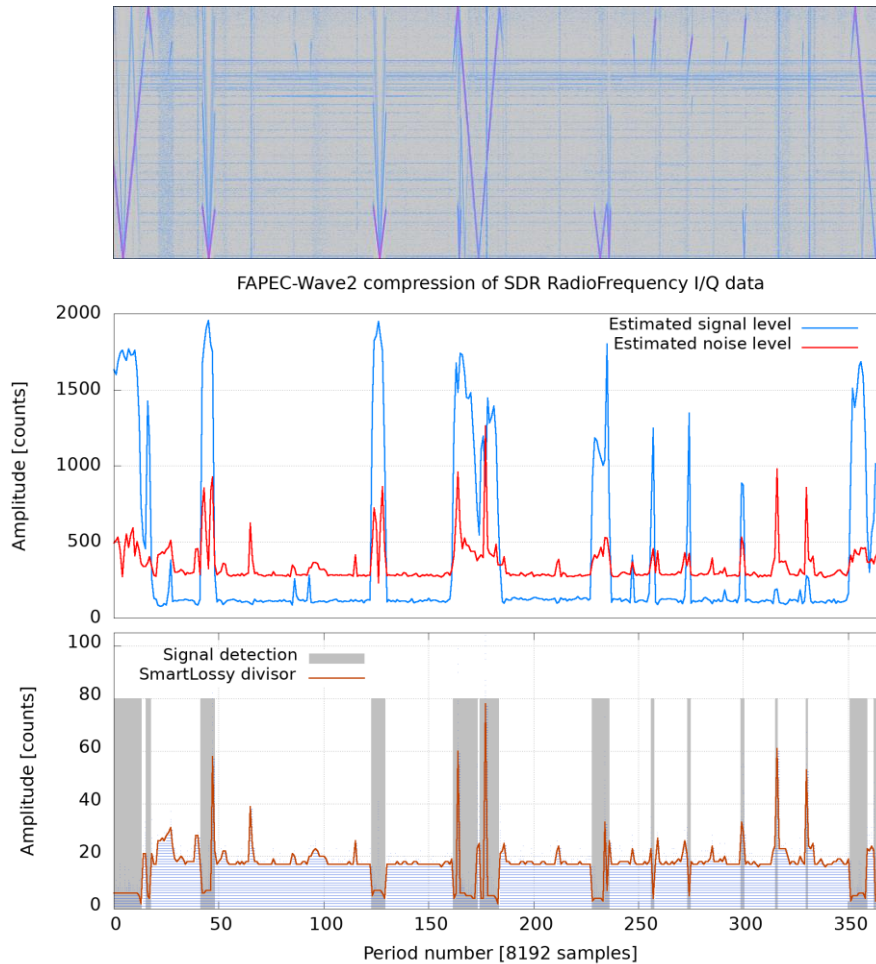


Fig. 5. Spectrogram (top), Levinson-Durbin estimations (middle) and Welch estimations (bottom) for an SDR file

We have also tested the near-lossless operation of Wave v2 on GNSS data, in order to evaluate its effect on this kind of spread spectrum signals. To do this, we have used the GNSS-SDR software package [23], configured with the kind help of its main developer to detect GPS and Galileo signals. On real OPS-SAT SDR files we were unable to get conclusive detections, probably due to their short duration, so we used a ground-based SDR data file with identical format. In this case, GNSS-SDR consistently detects Galileo signals. In lossless mode, FAPEC with its Wave v2 stage achieves a compression ratio of 1.8. When quantizing the raw input samples we can reach a ratio around 7 while still consistently detecting Galileo signals. It means that, with the adequate settings, we should be able to reliably use the smart lossy method even on data files with GNSS signals.

5. CONCLUSIONS AND FORTHCOMING WORK

In this work we have presented several improvements to the FAPEC image and radio-frequency data compression algorithms, called CILLIC and Wave, respectively. We have tested them on real in-orbit data from OPS-SAT. The main improvements achieved in CILLIC are the capability to reach higher compression ratios while still keeping a good image quality. It is now also able to generate thumbnails, as well as tiny text files with the outcome of a basic data analysis (focused on the fraction of flat blocks detected, for now). CILLIC is also able to compress video files, correctly detecting

the motion vectors, although the decorrelation algorithm in itself should be further improved. It also applies to the inter-band decorrelator. Regarding the Wave algorithm, we have slightly improved its lossless performance, and especially, we have implemented a smart lossy algorithm able to detect the presence (or absence) of signals in a data file. With this, we can significantly improve the ratios in RF files, and also generate tiny summary files with the signals detection outcome. These features can be used to optimize the download of image and RF files, prioritizing those for which FAPEC has detected relevant contents. In general, FAPEC confirms to be an excellent option for cubesats, owing to its versatility, ease of integration, basic data analysis capabilities, excellent performance and good ratios under nearly any situation.

REFERENCES

- [1] J. Puig-Suari, C. Turner and W. Ahlgren, "Development of the standard CubeSat deployer and a CubeSat class PicoSatellite," 2001 IEEE Aerospace Conference Proceedings (Cat. No.01TH8542), 2001, pp. 1/347-1/353 vol.1, doi: 10.1109/AERO.2001.931726.
- [2] O. Koudelka, D. Evans, "The OPS-SAT Nanosatellite Mission - A Flexible Platform for On-Board Experiments," Proc. of 11th IAA Symp. on Small Sat. for Earth Observation, Berlin, Germany, April 24-28, 2017, IAA-B11-0901
- [3] D. Evans, G. Labrèche, T. Mladenov, D. Marszk, V. Zelenevskiy et al. "OPS-SAT LEOP and Commissioning: Running a Nanosatellite Project in a Space Agency Context". <https://digitalcommons.usu.edu/smallsat/2022/all2022/113/>
- [4] G. Labrèche, D. Evans, D. Marszk, T. Mladenov, V. Shiradhonkar, V. Zelenevskiy. "Agile Development and Rapid Prototyping in a Flying Mission with Open-Source Software Reuse On-Board the OPS-SAT Spacecraft". <https://doi.org/10.2514/6.2022-0648>
- [5] G. Labrèche, D. Evans, D. Marszk, T. Mladenov, V. Shiradhonkar, V. Zelenevskiy. "Artificial Intelligence for Autonomous Planning and Scheduling of Image Acquisition with the SmartCam App On-Board the OPS-SAT Spacecraft". <https://arc.aiaa.org/doi/10.2514/6.2022-2508>
- [6] C. Coelho, D. Evans, O. Koudelka, "CCSDS Mission Operations Services on OPS-SAT," 10th IAA Symposium on Small Satellites for Earth Observation, Berlin, Germany, April 20-24, 2015, paper: IAA-B10-1301
- [7] J. Portell, R. Iudica, E. García-Berro et al. (2017) FAPEC, a versatile and efficient data compressor for space missions, IJRS 39:7, 2022-2042
- [8] J. Portell, R. Iudica, A. G. Villafranca, M. Hernández-Cabronero, I. Blanes, J. Serra-Sagristà (2020) CILLIC: Context Interpolation Lossless and Lossy Image Compressor, in ESA OBPDC VII
- [9] B. Artigues, J. Portell, A. G. Villafranca, H. Ahmadloo, E. García-Berro, "DWTF APEC: image data compression based on CCSDS 122.0 and fully adaptive prediction error coder." Journal of Appl. Remote Sens. 7 (1): 074592, 2013.
- [10] Stuart J. Russell, Peter Norvig (2003), Artificial Intelligence: A Modern Approach (2nd ed.), Upper Saddle River, New Jersey: Prentice Hall, pp. 111–114, ISBN 0-13-790395-2
- [11] H. Foroosh (Shekarforoush), J.B. Zerubia, and M. Berthod, "Extension of Phase Correlation to Subpixel Registration," IEEE Transactions on Image Processing, V. 11, No. 3, Mar. 2002, pp. 188-200.
- [12] J. Portell, M. Montón, R. Iudica, A. G. Villafranca (2020) FASEC: Fast And Simple Entropy Coder, ESA OBPDC VII
- [13] A. Martí (2021) Implementation of FAPEC decorrelation stages for IQ and water column data, Degree Thesis, UPC
- [14] P. Delsarte and Y. V. Genin (1986). "The split Levinson algorithm." IEEE Transactions on Acoustics, Speech, and Signal Processing, v. ASSP-34(3), pp. 470–478.
- [15] Josh Coalson. FLAC - format. 2001. url: <https://xiph.org/flac/format.html> (visited on 2022-09-20).
- [16] W. H. Jefferys, J. O. Berger (1991) "Sharpening Ockham's Razor On a Bayesian Strop".
- [17] Harry L. Van Trees. 2001. *Detection, Estimation, and Modulation Theory*. New York: Wiley.
- [18] Don Torrieri (2018). Principles of Spread-Spectrum Communication Systems, 4th ed.
- [19] Steven M. Kay. 1998. *Fundamentals of Statistical Signal Processing: Detection Theory*. Prentice-Hall PTR.
- [20] Sequeira, Samson, Rajas R. Mahajan, and Predrag Spasojević. 2012. "On the Noise Power Estimation in the Presence of the Signal for Energy-Based Sensing." In *2012 35th IEEE Sarnoff Symposium*, , 1–5.
- [21] P.D. Welch (1967), "The use of Fast Fourier Transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms" (PDF), IEEE Transactions on Audio and Electroacoustics, AU-15 (2): 70–73, Bibcode:1967ITAE...15...70W, doi:10.1109/TAU.1967.1161901
- [22] H. Akaike (1973), "Information theory and an extension of the maximum likelihood principle", in Petrov, B. N.; Csáki, F. (eds.), 2nd International Symposium on Information Theory, Tsahkadsor, Armenia, USSR, September 2-8, 1971, Budapest: Akadémiai Kiadó, pp. 267–281. Republished in Kotz, S.; Johnson, N. L., eds. (1992), Breakthroughs in Statistics, vol. I, Springer-Verlag, pp. 610–624.
- [23] C. Fernández-Prades, J. Arribas, P. Closas, C. Avilés, L. Esteve. (2011). GNSS-SDR: An open source tool for researchers and developers. 24th International Technical Meeting of the Satellite Division of the Institute of Navigation 2011, ION GNSS 2011. 2.