

# A VERSATILE IP CORE FOR REAL-TIME VIDEO COMPRESSION ON FUTURE ESA MISSIONS

Samuel Torres-Fau<sup>1</sup>, Felipe Machado<sup>1</sup>, Yubal Barrios<sup>1</sup>, Antonio J. Sánchez<sup>1</sup>, Luis Berrojo<sup>2</sup>, Roberto Sarmiento<sup>1</sup>, and Aniello Fiengo<sup>3</sup>

<sup>1</sup>*Institute for Applied Microelectronics (IUMA), University of Las Palmas de Gran Canaria (ULPGC)  
35017 Las Palmas de Gran Canaria, Spain*

<sup>2</sup>*Thales Alenia Space (TAS) in Spain  
28760 Tres Cantos, Madrid, Spain*

<sup>3</sup>*European Space Research and Technology Centre, European Space Agency  
2201 Noordwijk, The Netherlands*

## ABSTRACT

This paper presents the design, development and implementation of a versatile H264 video encoder IP core tailored for space applications. Given the wide range of space applications — such as Earth Observation, object tracking, and navigation assistance — the developed architecture can be easily customized to their specific requirements. Starting from a base intra-prediction architecture, a variety of options can be added, which include chroma processing, inter-prediction, constant bitrate operational mode, external memory usage and run-time configuration. The IP is capable of autonomously processing video sequences, but an optional co-processor may also interact with it through standard bus interfaces. The hybrid DSE analyzed the trade-off between performance and complexity in order to optimize the design process and develop an IP that meets the project requirements, and the hierarchical, iterative development process has ultimately resulted in the final design. The synthesis and implementation results have been obtained targeting the Xilinx Kintex Ultrascale KU040 FPGA. These results demonstrate an efficient resource utilization while maintaining robust compression performance when the Intellectual Property (IP) is configured with all the available extensions, while a low-complexity version for resource-constrained applications is also available by selecting the base configuration.

## 1. INTRODUCTION

Space missions have increasingly included more higher resolution optical imaging sensors in recent years. This trend allows space systems to capture larger amounts of various types of data [1]. This is the case for multispectral and hyperspectral imaging sensors, which allow to analyze numerous narrow spectral channels, and for the higher spatial resolution of the newer 2-dimensional optical cameras. Furthermore, capturing video on space enables temporal analysis for various relevant applications, which include Earth Observation [2, 3], object tracking [4], navigation aid [5] and landing assistance systems [6].

The substantial volume of data produced by modern sensors requires implementing on-board compression techniques. Compression is essential to efficiently handle and store the captured information. This requirement is particularly critical on space systems, which are subject to stringent constraints related to hardware resources, power consumption and transmission link bandwidth. Although there are generic data compression standards, such as the Consultative Committee for Space Data Systems (CCSDS) 121.0-B-3, the spatio-temporal dependencies of video sequences often require domain specific solutions to improve compression ratios. Besides, video cameras can generate raw data at throughputs of gigabits per second, making impractical to store the raw information on-board or to transmit it to ground using available communication links. The H.264 standard (hereafter H264) [7], also known as Advanced Video Coding (AVC) or MPEG-4 Part 10, is a well-established and widely proven video industry standard for video coding that allows to achieve high compression ratios while keeping quality losses at reasonable and controlled levels. Originally published in August 2004, H264 introduced significant improvements over previous video compression standards, whilst allowing to keep the computational costs at reasonable levels.

This work introduces the outcome of the *Efficient Video Compression for Space* project, funded by the European Space Agency (ESA) under the contract No. AO/1-1-10954/21/NL/MGu and led by Thales Alenia Space in Spain (TASiS). The main objective of this project is the development of an H264 compliant, technology-agnostic video compression IP Core, followed by its verification, validation and integration in the ESA IP Core portfolio. IUMA is in charge of the development and verification, while TASiS is responsible for the validation on a space-representative technology. This IP core is expected to be used on forthcoming ESA missions that

include on-board video acquisition and processing systems. The IP includes a streaming-oriented control for standalone operation, while also being easily controllable from an external unit, such as a CPU or a SoC, thanks to its compatibility with standard bus protocols (e.g., Advanced eXtensible Interface (AXI)4 [8]). The IP Core includes many configurations ranging from low-complexity to high-performance in order to adapt it to a wider range of applications, while enabling further refinement through a set of available system parameters.

The rest of the paper is structured as follows: Section 2 provides an overview of the H264 compression standard. Section 3 details the development process conducted for the *Efficient Video Compression for Space* project. Next, section 4 summarizes the implemented video compression IP Core, providing further insights on the structural adaptability of the design. Section 5 presents the hardware occupancy and performance results, which are also compared against other commercial H264 hardware implementations. Finally, the main conclusions of this work are drawn in Section 6.

## 2. H264 OVERVIEW

Several approaches exist for developing video compression solutions, although just a few are well-established standards with proven effectiveness across various scenarios. One prominent example is H264 [7], a video compression standard developed by the Joint Video Team (JVT). H264 achieves high-quality video at lower bit rates than previous standards, while maintaining computational complexity at manageable levels. This combination has led to its widespread adoption across a diverse array of applications during the last two decades.

Similar to other compression standards, an H264 encoder can be divided into two main components: Prediction and Encoding. These stages operate on each individual macroblock within a frame. H264 partitions frames into macroblocks, which are blocks of 16x16 samples of the luma component and their corresponding chroma components. All the macroblocks within a frame are predicted and encoded line-by-line, in raster order.

### 2.1. Prediction

Video compression exploits the spatial and temporal redundancies to predict the value of the pixels in order to reduce the encoded video size.

Spatial redundancies are used to predict pixel values from already processed pixels within the same frame, therefore, it is often called **intra-prediction** because the prediction does not use information outside of the current frame. Frames that have been predicted solely using intra-prediction are called I-Frames. In the case of H264, intra-prediction mainly uses two block sizes: either a whole macroblock (hereafter *Intra-16 prediction*) or 4x4 blocks (henceforth named *Intra-4 prediction*). Both have different prediction modes depending on which neighbouring pixels and prediction algorithm have been selected. Intra-16 prediction has four prediction modes (Vertical, Horizontal, DC and Plane), while Intra-4 has nine prediction modes (Vertical, Horizontal, DC and six diagonals). Using higher granularity, such as in Intra-4, produces better predictions at the cost of including more prediction information for each of the sixteen 4x4 blocks of a macroblock. Therefore, Intra-16 prediction tends to suit better when large areas of the frame have similar pixel values (as in uniform backgrounds), while Intra-4 usually works better in areas with more detail.

Temporal redundancies can be used to predict future frames from previously processed frames. H264 allows predicting motion between blocks of different frames in order to achieve better estimations (also called **inter-prediction**). Frames that are predicted from previous frames are called P-Frames.

Inter-prediction requires more resources, because a whole frame needs to be stored, and motion estimation is cost-intensive; however, they usually get much better compression ratios than I-frames.

I-Frames are always necessary to have a frame that can be independently visualized. Consequently, a Group Of Pictures (henceforth named GOP) refers to a sequence of frames that starts with an I-Frame and defines the order of subsequent frames that are temporally predicted. There are many GOP structures, because H264 also considers B-Frames, which are predicted from previous and future frames, but one of the simplest GOP structures is to have just an I-Frame followed by a number of P-Frames, in which each P-Frame is predicted from the immediate previous frame.

After the prediction, the resulting blocks with the residuals (differences) are transformed and quantized. The quantization can further reduce the size and number of coefficients sent to the encoding subsystem. The mag-

nitude of the quantization is determined by Quantization Parameter (QP), an integer whose value ranges from 0 (highest quality) to 51 (lowest quality).

## 2.2. Encoding

The H264 encoding process uses an adapted method to compress video data efficiently. After the prediction stage, the remaining transformed and quantized residual data of each macroblock is processed with an entropy coding method. H264 standard defines two alternatives, as it will be further explained. This step is essential for substantially reducing the data required for storage or transmission.

Each encoded macroblock consists of two differentiated components: the macroblock header and the residual data. The macroblock header contains a set of syntax elements such as the prediction type, the applied quantization parameter value, motion vectors, and other relevant metadata. These metadata are crucial for the correct decompression of the corresponding pixels, and they are generally encoded using Golomb-Exponential coding, a specific type of universal codes for unsigned values (signed values are also supported by adding an extended mapping). This encoding technique assumes that lower values are more likely to occur, thus assigning them shorter codes.

Regarding the residual encoding, H264 proposes variable-length coding and arithmetic coding techniques to maximize compression ratios, which ultimately lead to smaller bitstreams. The two defined encoders are Context-Adaptive Variable Length Coding (CAVLC) and Context-Adaptive Binary Arithmetic Coding (CABAC). Both methods dynamically adjust the coding based on the statistical characteristics of the data, allowing for more efficient representations. All the H264 decoders must support CAVLC-coded bitstreams, while just the high-end H264 decoders are able to decode bitstreams generated with CABAC. CABAC enhances the compression ratio by employing arithmetic coding and advanced context model selection, though these gains are accompanied by a substantial increase in coding complexity [9].

The encoded information is organized by following a layered scheme, namely Network Abstraction Layer (NAL), which is suitable for properly storing, transmitting and decoding it.

## 3. METHODOLOGY

The objective of the *Efficient Video Compression for Space* project is the development of the first H264 Encoder IP Core in the ESA IP portfolio. This project has been conducted in a series of well-defined stages. An initial Design Space Exploration (DSE) was performed using software profiling tools as well as the High-Level Synthesis (HLS) methodology [10], which enabled the effective analysis of the trade-offs associated with the inclusion of key components of the H264 compression standard. The profiling tools were executed over the H264 reference implementation, the JM Software [11]. Once decided which elements would be included in the IP, various iterative cycles for design, integration and verification of the system have been performed. The completion of each of these cycles has led to certain milestones, starting from a minimal working implementation of the encoder and leading to the full encoder IP core with all its configuration capabilities. For example, the first cycle led to the development and verification of a basic core that supported exclusively Intra 4x4 prediction modes, while the second iteration resulted in a version that also included Intra 16x16 modes.

### 3.1. Design Space Exploration

H264 is a complex standard that includes many features that may drastically increase complexity but not always offer a significant compression ratio improvement. Under this context, DSE has proven crucial in order to identify a convenient set of features to be implemented on the final design. Considering the analyzed dataset, the following key decisions were obtained from the DSE:

- Support for I-Frames and P-Frames. The study demonstrated that the P-frames offer significant compression ratios over I-frames. However, B-Frames slightly improved the compression rate over P-frames for the studied representative scenarios, while increasing the design complexity (both in computational power and memory requirements) to unreasonable levels.

- The usage of Rate Distortion Optimized (RDO) algorithm for the implementation of the rate control module was also discarded based on the same reason as for B-frames.
- The impact of the inclusion of the different Intra-prediction modes was thoroughly evaluated, aiding the decision-making process regarding which modes to include.
- Various Motion Estimation (ME) algorithms were analyzed. Results showed that complex algorithms such as the Full Search (FS) did not offer significant gains over simpler ones, concluding that the algorithm to be included shall be as simple as possible. It also helped to limit the search range due to the same reason.
- Sum of Absolute Differences (SAD) error metric was selected over other choices for its good balance between performance and computational complexity.

### 3.2. Incremental development

The DSE defined which characteristics of the H264 to incorporate in the IP, paving the way for the design phase to commence shortly thereafter. This phase followed an iterative process. Partial yet correct versions of the compressor core were deployed before the final version. The two main functional subsystems of the design, the prediction and encoding subsystems, were designed independently, with functionality divided into smaller components. This approach ensured a proper integration between the components and reduced design and verification times. A general representation of the development flow is shown at Figure 1. Each component was individually described and verified, and consequently integrated in its corresponding subsystems. The same approach was used for the subsystems, which were ultimately integrated in the top-level module. In verification campaign, which has been a three-level process, a modified version of the JM Software [11] has been extensively used to generate test vectors for component-, subsystem-, and system-level verification runs. The hierarchical verification process facilitated efficient bug identification and minimised errors in regression tests.

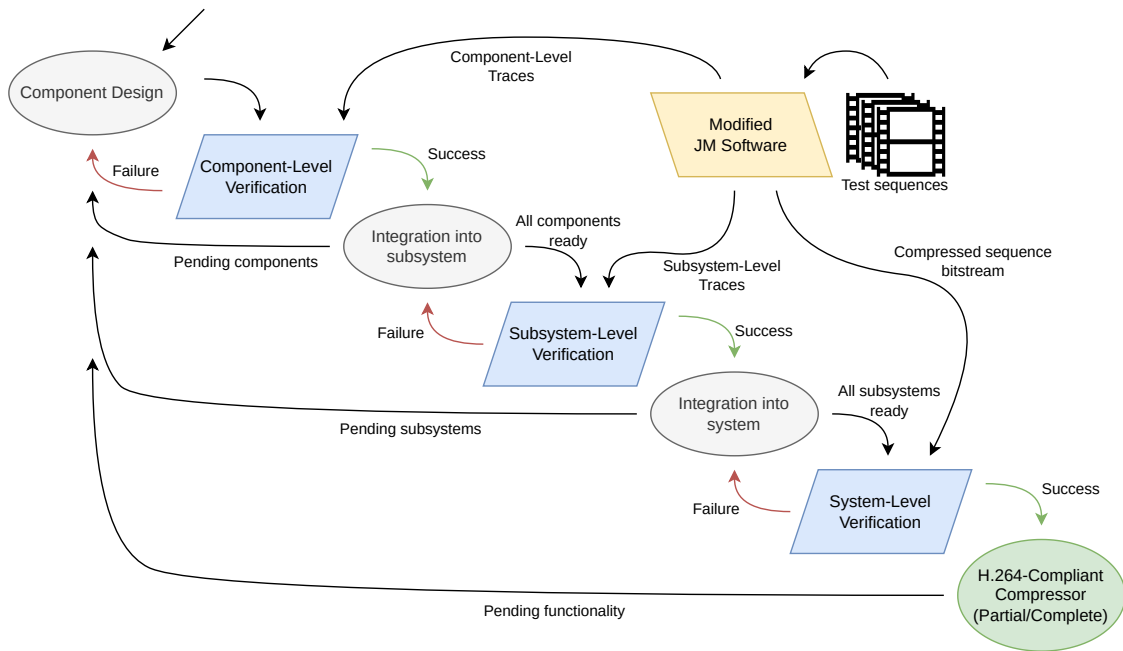


Figure 1. Design & verification flow diagram

## 4. VERSATILE ARCHITECTURE

The IP architecture is composed by the two main subsystems of the H264 standard: prediction (§2.1) and encoding (§2.2). There is a third optional component that allows configuring some of the encoder parameters at runtime.

The architecture is parametric, allowing implementing different configurations depending on the system requirements imposed by the input video characteristics, hardware utilization and compression performance (quality and compression ratio). The minimum implementation is oriented for resource-scarce devices. This configuration only uses intra-prediction without rate-control, does not allow to modify the internal registers (i.e., runtime configuration disabled, system parameters fixed) and is intended for monochrome video only. The complexity and the performance of the implemented IP core will increase if the following options are incorporated:

1. Chroma 4:2:0 format: it implements the necessary prediction and encoding blocks to encode 4:2:0 video.
2. Rate Control: includes a block to adjust the video quality (QP), to get the desired compression ratio.
3. Inter-prediction: includes the prediction modules to predict the current pixels from the previous frame.
4. Frame buffer allocation: in case that inter-prediction is selected, the memory to save the previous frame can be located within the IP core or outside.
5. IP runtime configuration: some of the IP internal registers can be modified through AXI/AHB bus. The registers that can be modified are: target compression rate (if rate control has been selected), the initial QP and its higher and lower limits, and the GOP (if inter-prediction has been selected).

Figure 2 reflects the IP core block diagram with the different implementation options. Note the simplicity of the minimum configuration, which only includes the white blocks.

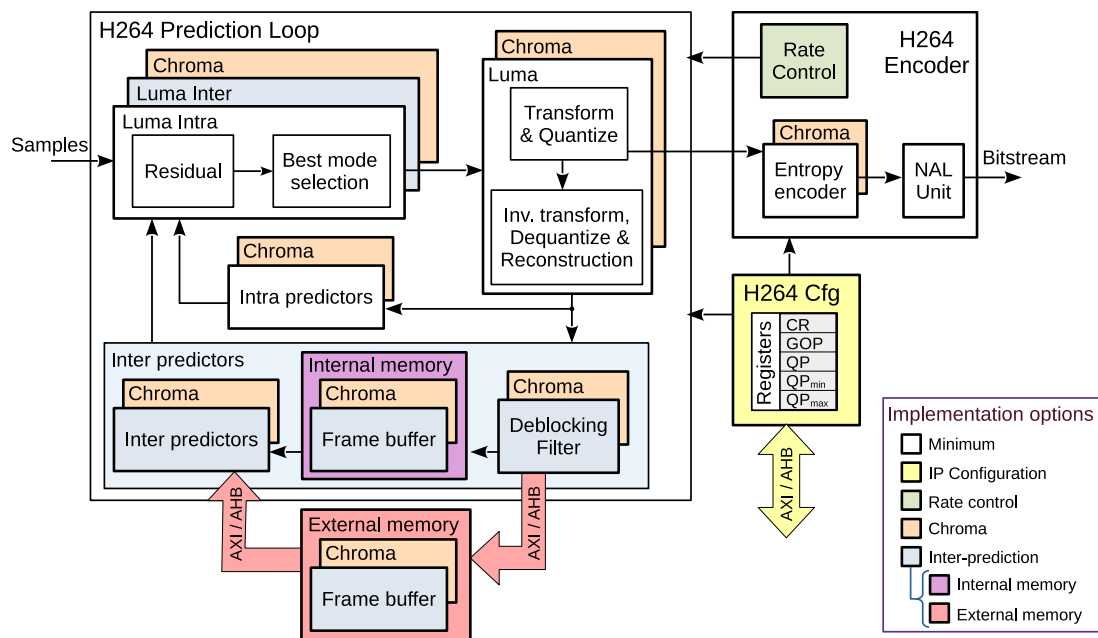


Figure 2. Simplified block diagram with the implementation options of the IP core

There are other implementation options, such as the frame size, the implementation of the EDAC mechanism in certain critical memories of the system and the type of the external bus interface (AXI or Advanced High-performance Bus (AHB)). External bus interfaces can be used to accomplish two different purposes: if runtime configuration is enabled, the IP registers can be accessed; on the other hand, the prediction block may employ an external memory to store the reconstructed reference frame needed for inter-prediction, to notably reduce the internal memory resources. In addition, a quasi-lossless mode can also be set by fixing the QP to 0.

The IP data interfaces are driven by standard ready/valid handshaking, as shown in 3, which simplifies the integration of the encoder in data driven systems. Two supplementary flags, *EOF* and *EOG*, are provided to indicate the completion of a frame or a GOP, respectively.

The IP has been widely verified using both small, simple synthetic videos as well as a set of sequences that represent various scenarios for typical use-cases for the IP (different space footages). The adopted verification methodology required to run multiple regression tests in order to assure that the design behaves as expected.

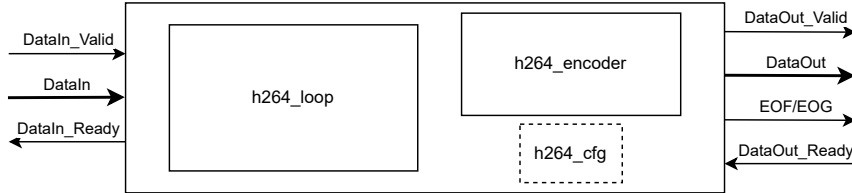


Figure 3. Data interfaces of the IP video encoder

## 5. RESULTS

The developed video compression IP core is able to autonomously process sequences of images generating H264-compliant bitstreams. Once the system was completely verified, synthesis and place & route were performed for the target technology, which was the Xilinx Kintex UltraScale KU040 FPGA. This device is technologically equivalent to the space-grade, radiant-tolerant Xilinx XQRKU060 FPGA. Table 1 shows the implementation results in terms of maximum clock frequency and resource occupation for different IP core configurations. These results are for compile-time version using an AXI interface for external memory access.

	<b>Intra-only Monochrome</b>	<b>Intra-only + Color</b>	<b>Inter Monochrome</b>	<b>Inter + Color</b>
LUT	11803 (4.8%)	19626 (8.1%)	42199 (17.4%)	56484 (23.3%)
FF	23065 (4.8%)	32331 (6.7%)	85146 (17.5%)	101756 (21.0%)
BRAM	0.5 (0.1%)	0.5 (0.1%)	32.5 (5.4%)	44.5 (7.4%)
DSP	16 (0.8%)	18 (0.9%)	16 (0.8%)	18 (0.9%)

Table 1. Implementation results on Xilinx Kintex Ultrascale XKCU040

The usage of hardware resources for the basic configuration demonstrates that it is an ideal choice for tightly hardware-constrained systems given its low Look-Up Table (LUT) and Flip-Flop (FF) usage, and the negligible Block RAM (BRAM) and Digital Signal Processor (DSP) utilization. If inter-prediction is included both the LUT and FF utilization increase, due to the inclusion of an additional prediction loop with independent processing units. In addition, 32 BRAMs are also required, mainly to the inclusion of a motion estimator and the deblocking filter. Incorporating color increases the number of used LUTs and memory elements due to the chroma-specific components

The performance has been measured under various typical configurations, and it is summarized in Table 2. The obtained performance metrics depend on the type of frame being encoded, as well as on the properties of the sequence (frame size, movement type, chroma format) and the compression parameters (constant quality or constant bitrate, GOP size).

The developed architecture is capable of handling the typical framerates of the cameras included in many space missions, as it supports 720p@45 fps, 1080p@20 fps and 2160p@5 fps.

<b>Frame-type</b>	<b>Mean Performance (cycles/macroblock)</b>	<b>Mean Performance (Mpixel/s)</b>	<b>FullHD Framerate (frames/second)</b>
Intra Chroma 4:2:0	1000	35.84	17.3
Inter Chroma 4:2:0	775	46.25	22.3
Intra Monochrome	800	44.80	21.6
Inter Monochrome	650	55.14	26.6

Table 2. Performance results on Xilinx Kintex Ultrascale XKCU040

Finally, a comparison against other commercial H264-based compression solutions is offered in Table 3. As it can be observed, the developed architecture offers the lowest LUT usage as well as a negligible DSP usage. Regarding the intra-only architecture, it additionally offers nearly-zero BRAM usage. It must be remarked that even lower occupation rates may be achieved by setting monochrome processing, as it was previously shown in Table 1. One downside that the comparison shows is that our IP has a slightly lower performance when compared to the other encoders. Nevertheless, internal studies have demonstrated that it is possible to fine-tune the pipelined architecture to achieve greater performance, if required.

	<b>This work</b> (Inter+Intra)	<b>This work</b> (Intra-only)	<b>xkAVC</b> <b>k3.pe</b> [12] (Inter+Intra)	<b>xkAVC</b> <b>k3.ie</b> [12] (Intra-only)	SOC H264 HD Encoder [13]	SOC H264 HD Encoder [13] (Intra-Only)
Max. clock freq. (MHz)	140		120		-	
Platform	Xilinx KU040		Xilinx ZC102		Xilinx Artix/Kintex/Ultrascale	
Framerate	1080p@20	1080p@20	1080p@30	1080p@60	1080p@60	1080p@60
LUT	61009	23769	166557	52414	~235000	~45000
FF	106997	37345	73099	30919	-	-
BRAM	44.5	0.5	19	9.5	~280(10Mb)	~110 (4 Mb)
DSP	18	18	185	112	235	140

Table 3. Comparison against other available commercial solutions

## 6. CONCLUSIONS

The first H264-compliant Video Encoder of the ESA portfolio has been presented. Its development has been conducted in well-defined phases. The initial DSE analysis has been essential to establish which features of the H264 standard had to be included in the system. The iterative design methodology has allowed to periodically generate compressor solutions that have progressively integrated more complex features. It also has enabled different team members to work in parallel, thus shortening development times.

Our solution is a highly-configurable H264 video compressor. The High-Performance implementation option offers the greater compression performance, while the Low-Complexity option yields the lowest hardware occupation metrics. An even lower-complexity architecture is possible by setting the monochrome flag. This architectural flexibility aims to ease the integration of the IP Core in the potential missions that require including an on-board video compression solution. In the same way, the usage of standard interfaces also eases the integration process.

Finally, the achieved results demonstrate the flexibility of the design for space-representative technologies. Next step is a physical validation in a space-representative technology by Thales Alenia Space in Spain, which is expected to start soon.

## REFERENCES

1. Y. O. Ouma, “Advancements in medium and high resolution Earth observation for land-surface imaging: Evolutions, future trends and contributions to sustainable development,” *Advances in Space Research*, vol. 57, no. 1, pp. 110–126, 2016.
2. Y. Barrios, F. Sanjuan, G. Bordot, H. Sharif, J. Bernier, and S. López, “Demonstrator Development of a Next-Generation Video Instrument for Earth Observation,” in *2023 26th Euromicro Conference on Digital System Design (DSD)*, 2023, pp. 407–414.
3. G. Kopsiaftis and K. Karantzalos, “Vehicle detection and traffic density monitoring from very high resolution satellite video data,” in *2015 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, 2015, pp. 1881–1884.
4. B. Du, S. Cai, and C. Wu, “Object Tracking in Satellite Videos Based on a Multiframe Optical Flow Tracker,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 8, pp. 3043–3055, 2019.
5. R. K. Rangel, K. H. Kienitz, and M. P. Brandao, “Development of a complete UAV system using COTS equipment,” in *2009 IEEE Aerospace conference*, 2009, pp. 1–11.
6. J. Vezinet, A.-C. Escher, A. Guillet, and C. Macabiau, “State of the art of image-aided navigation techniques for aircraft approach and landing,” in *ION ITM 2013, International Technical Meeting of The Institute of Navigation*, San Diego, United States, Jan. 2013, pp. pp 473–607. [Online]. Available: <https://enac.hal.science/hal-01022434>
7. Telecommunication Standardization Sector of International Telecommunication Union, “Recommendation ITU-T H.264: Advanced video coding for generic audiovisual services,” Tech. Rep., 8 2021. [Online]. Available: <https://www.itu.int/rec/T-REC-H.264>
8. arm, *AMBA AXI4 Protocol Specification*, 9 2023, issue K, <https://developer.arm.com/documentation/ih0022/k/>.
9. I. E. Richardson, *The H.264 Advanced Video Compression Standard*. Wiley, 2010.
10. F. Machado, Y. Barrios, R. Sarmiento, F. Sanjuan, and A. Fiengo, “A Hardware/Software Design Space Exploration for Efficient Video Compression on ESA missions,” in *2023 European Data Handling & Data Processing Conference (EDHPC)*, 2023.
11. Telecommunication Standardization Sector of International Telecommunication Union, “H.264.2: Reference software for ITU-T H.264 advanced video coding,” International Telecommunication Union, Tech. Rep., 2 2016. [Online]. Available: <https://www.itu.int/rec/T-REC-H.264.2>
12. XK SILICON, “XK - xkAVC Encoder for FPGA (K3).” [Online]. Available: <http://xk.openasic.org/product/3>
13. SOC Technologies, “H.264 HD Video Encoder IP Core.” [Online]. Available: <http://www.soctechnologies.com/ip-cores/ip-core-h264-encoder>