JPEG XS – A new standard for visually lossless low-latency lightweight image coding

Antonin Descampe, Thomas Richter, Touradj Ebrahimi, Siegfried Foessel, Joachim Keinert, Tim Bruylants, Pascal Pellegrin, Charles Buysschaert, and Gaël Rouvroy.

Abstract

JPEG XS is a new International Standard from the JPEG Committee (formally known as ISO/IEC JTC1/SC29/WG1). It defines an interoperable, visually lossless low-latency lightweight image coding that can be used for mezzanine compression within any AV market. Among the targeted use cases, one can cite video transport over professional video links (SDI, IP, Ethernet), real-time video storage, memory buffers, omnidirectional video capture and rendering, and sensor compression (for example in cameras and in the automotive industry). The Core Coding System is composed of an optional color transform, a wavelet transform and a novel entropy encoder, processing groups of coefficients by coding their magnitude level and packing the magnitude refinement. Such a design allows for visually transparent quality at moderate compression ratios, scalable end-to-end latency that ranges from less than one line to a maximum of 32 lines of the image, and a low complexity real-time implementation in ASIC, FPGA, CPU and GPU. This paper details the key features of this new standard and the profiles and formats that have been defined so far for the various applications. It also gives a technical description of the Core Coding System. Finally, the latest performance evaluation results of recent implementations of the standard are presented, followed by the current status of the ongoing standardization process and future milestones.

Index Terms

codec, video, image, standard, uncompressed, broadcast, sensor compression,

I. Introduction

VIDEO bandwidth requirements are growing fast, as video resolutions, frame rates and the amount of streams to manage are constantly increasing. The capacity of video links and communication channels is growing too, but at a slower pace, and more importantly, only through huge investments requiring amortization over several years. The broadcast industry illustrates this situation very well: for the transport of video streams, manufacturers are currently still transitioning from HD-SDI (1.5 Gbps, supporting 1080i60) to 3G-SDI (3 Gbps, supporting 1080p60), while UHD-4k is already being deployed — requiring 12G-SDI or 10 Gbps Ethernet — and UHD-8k is being explored.

As a consequence, uncompressed storage and live video transmission becomes unaffordable and unmanageable within the currently deployed systems and infrastructures, while next generation channels are still being tested or are still too expensive. Given this reality, the use of a lightweight compression scheme is very attractive, as it allows for a smooth and long-lived transition between successive generations of infrastructures and protocols. It also increases flexibility in the management of video streams, i.e. it allows multiplexing of various streams onto one single link, or it is able to carry a single stream on a small capacity link, usually also allowing longer cable runs.

Such a lightweight compression scheme should be developed in such a way that it allows for increasing resolution, frame rate and number of streams while safeguarding all the advantages of an uncompressed stream, i.e. interoperability, transparent quality, low power consumption, low latency in coding and decoding, ease of implementation, small size on chip and fast software running on a general purpose CPU. One might think that such a compression scheme can be found among the numerous existing video compression standards. However, in practice, existing standards such as JPEG, JPEG 2000, HEVC, ProRes, DSC, VDC-M, or VC-2 do not fit, as none of them meets all of the above requirements.

In this context, the JPEG Committee (formally known as ISO/IEC SC29 WG1) has standardized a novel compression codec called JPEG XS. Improving the rate-distortion trade-off has often been the highest priority of previous standardization initiatives while complexity and latency have been, at best, only secondary goals. On the other hand, the JPEG XS project has been focused from the very beginning on providing the best trade-off between quality, complexity and latency for the targeted use cases.

This paper is organized as follows: after the introduction in this section, Sections II and III present the JPEG XS targeted applications and use cases, and derive from them the required key features of this new codec. Section IV shows that no existing codec meets all of the identified requirements. Then, a technical overview of the Core Coding System follows in Section V. A break-down of the various sources of latency in the codec is given in section VI, after which application profiles and formats

A. Descampe is with Université catholique de Louvain (UCLouvain), Belgium.
T. Bruylants, C. Buysschaert, P. Pellegrin and G. Rouvroy are with intoPIX SA, Belgium.
T. Ebrahimi is with Ecole Polytechnique Fédérale de Lausanne, Switzerland.
S. Foessel, J. Keinert and T. Richter are with Fraunhofer IIS, Germany.
This is the final, peer-reviewed version of this paper published in Proceedings of the IEEE (DOI 10.1109/JPROC.2021.3080916).

TABLE I
TYPICAL VIDEO FORMATS AND COMPRESSION RATIOS REQUIRED TO TRANSPORT THEM ON THE COMMON INTERFACES.

Format and bit-rate	HD-SDI	3G-SDI	12G-SDI	1G Ethernet	10G Ethernet
HD/2k 60p/422/10 bits 2.7 Gbps	1x ∼2:1 2x ∼4:1	1x uncompr.	1x uncompr.	1x ∼3:1 2x ∼6:1	3 x uncompr.
HD/2k 60p/444/12 bits 4.8 Gbps	1x ∼3.5:1 2x ∼4:1	1x ∼2:1 2x ∼4:1	1x uncompr.	1x ∼6:1	2 x uncompr.
UHD-1/4k 60p/422/10 bits 10.8 Gbps	1x ∼8:1	1x ∼4:1	1x uncompr.	1x ∼12:1	3x ∼4:1
UHD-1/4k 60p/444/12 bits 19 Gbps	-	1x ∼7:1	1x ∼2:1 2x ∼4:1	-	1x ∼2:1 2x ∼4:1
UHD-2/8k 120p/422/10 bits 85 Gbps	-	-	1x ∼8:1	-	1x ∼10:1

are presented in Section VII. Finally, section VIII summarizes the latest available assessments of performance, and Section IX gives an overview of the current status of the standardization process and introduces some upcoming extensions.

II. USE CASES

In a nutshell, JPEG XS is a candidate technology wherever uncompressed video is used today. However, it has been specifically designed to meet the requirements of live production, broadcast and digital cinema workflows, Pro-AV markets, Virtual Reality (VR) gaming, and image sensor compression [1].

- **Transport over video links and IP networks**. In broadcast studios the massively deployed SDI infrastructures (mainly HD-SDI and 3G-SDI) are gradually being replaced by IP-based networks. Today, the preference is to use 1 Gigabit Ethernet [GE] links for remote production and 10 GE infrastructures for in-house studios. 25, 40 or 100 GE links are not usually affordable yet. So, given the available bandwidth, uncompressed video is not longer viable, as 4K, 8K, and higher frame rates also need to be supported. This means that lightweight compression which preserves the visual quality can be a good solution for transporting such video streams over existing and new infrastructures. Table I presents typical video formats with compression ratio indications for common interfaces. As shown, with compression ratios up to 12:1, nearly all cases are covered. In addition, robustness to multiple encoding and decoding cycles is also critical when chaining multiple devices that each re-compress the signal. Previous research [2] recommends that at least 7 compression-decompression cycles should be supported. As far as latency is concerned, for a complete live production chain the total video latency must be no more than 80-100 ms including all inherent sources of latency (codec, jitter buffering, processing delay, redundancy buffering, etc) [3], [4]. In real-time video systems in LAN networking (professional or prosumer), most of video-over-IP systems are asynchronous, meaning that receiver and transmitter are not synchronized. In such environment, in the case of underflow (i.e. data arriving too late), the decoder will typically repeat the previous frame. The consequence is that the latency will then incrementally jump by 16 ms (at 60Hz). Therefore, implementers aim to keep any additional source of latency (such as several encoding-decoding cycles) as low as possible to avoid further increasing the latency and possibly trigger human-perceptible delay [5]. Based on these use cases, the codec-related latency should remain less than a few dozens of lines.
- **Real-time video storage**. Similar to the transport of video streams, storage of high-resolution streams requires lightweight compression to allow real-time writing to lower cost, and thus inherently slower, storage devices such the SD cards found in cameras. Moreover, multiple encoding-decoding cycles should not impair the image quality.
- **Frame buffer compression**. Lightweight compression of buffers inside video processing devices can drastically reduce the system's form factor, decrease the number of interconnect wires and extend battery life for battery powered systems. For instance, JPEG XS could be used in high refresh-rate (+120 Hz) display buffers, storage and replay buffers for high speed cameras, or reference frame buffers in AVC/H.264 or HEVC/H.265 hardware codecs [6].
- **Omnidirectional video capture and rendering**. JPEG XS is also intended to be used in head-mounted displays for Virtual or Augmented Reality (VR/AR). To get an immersive experience, displays with resolutions greater than 8 Megapixels and 90 frames per second and per eye are necessary. Such applications require a very low latency coding scheme so as to ensure tight synchronization between movement and display.
- **Sensor compression**. More and more image sensors are used in industrial environments with increased resolution. In this context, JPEG XS offers a convenient way to ensure transport of image sequences within industrial networks by supporting direct Bayer pattern compression of image sensor data. This allows the sensor image data to be processed with maximum responsiveness and very low latency along the whole data flow. In terms of implementation and given the number of sensors, power consumption needs to be constrained as much as possible because of thermal considerations and the necessary operation in all kinds of climatic conditions.

III. Key features

Based on the above-described use cases, the following requirements have been identified, and are now the key features of JPEG XS [1].

- **Visually lossless quality**. The difference between the original image or image sequence and the same image or image sequence after compression and decompression must be undetectable by a human observer under normal viewing conditions. Corresponding compression ratios required to achieve this quality range from 2:1 to 10:1 for both 4:4:4 and 4:2:2 images with up to 12-bit component precision but can also be higher depending on the nature of the image, or on the requirements of the targeted application. For certain use cases, even higher quality is required, such as flicker-resilient visually lossless picture quality (as defined in [7]). The corresponding compression ratios required to achieve this quality range from 2:1 to 6:1. To ensure JPEG XS achieves such quality, a specific set of subjective evaluation procedures was used during the technology selection and core experiments of the project, as presented in Section VIII-C and detailed in [8].
- **Multi-generation robustness**, i.e. no significant quality degradation for up to 10 encoding-decoding cycles.
- **Multi-platform interoperability**. The JPEG XS use cases require real-time implementations on several different platforms: CPU, GPU, FPGA and ASIC. Each of these platforms is best exploited when a specific degree of parallelism is available in the implemented codec. For instance, a multi-core CPU implementation will benefit from a coarse-grained parallelism while GPU or FPGA will better take advantage of a fine-grained parallelism. Therefore, in order to support optimally the different target platforms, the JPEG XS codec allows for different kinds of end-to-end parallelization. More importantly, real-time encoding on a given platform (a FPGA for instance, exploiting a fine-grained parallelism) allows real-time decoding of the generated codestream on any other platform (including for instance a multi-core CPU exploiting a different kind of parallelism), sacrificing neither the low complexity nor the low latency features described below.
- **Low complexity**, both in hardware and software. For JPEG XS to be a legitimate candidate to replace uncompressed video transport, implementations of very low complexity need to be achievable. In practice, as far as software is concerned, JPEG XS has been designed so that a Core i7 processor is able to process 4k 4:4:4 60p content in real time. As regards hardware, FPGA implementations do not require any external memory and do not occupy more than 50% of Artix7 XC7A200T or 25% of a Cyclone V 5CEA9 when applied to 4k 4:4:4 60p content.
- **Low latency**. As indicated above, whether it be in video transport applications (especially live production), in AR/VR applications, or in any other use case requiring tight synchronization between the signal and a human interaction, the cumulative delay caused by all the processing steps undergone by the signal has to be kept below the threshold of human perception. To this end, based on inputs from the different application fields and specifications discussed within professional associations like AIMS[1], JPEG XS requirements targeted a maximum algorithmic end-to-end latency of 32 lines. It eventually offers a scalable algorithmic latency, ranging from a small number of lines down to less than a single line for a combined encoder-decoder suite. As further described in Section VI, depending on the use case and the targeted platform, the total latency (including both the algorithmic latency and the implementation latency) might be higher. This is particularly the case in CPU and GPU implementations. It should be noted that the low latency requirement also serves the low complexity requirement: in a hardware FPGA implementation, the amount of memory blocks is critical. Even with an end-to-end latency of 32 lines, the percentage of memory blocks used is usually 2 or 3 times bigger than the percentage of logic elements. Therefore, targeting a lower amount of lines is a must in some markets in order to fit in low-end FPGA chips. The same is true for ASIC implementations where for an encoder+decoder system having an end-to-end latency of 16 lines, the memory alone will already take up more than 90% of the chip size. This also explains why a scalable algorithmic latency has eventually been chosen for JPEG XS.

IV. Comparison with other codecs

Based on the key features described above, it is easy to see that no pre-existing standard combines them all. JPEG LS [9] and JPEG [10] as well as its successor JPEG XT [11], which provides backward compatible support of higher bit depths, make precise rate control difficult, and nor do they allow latency to be limited.

JPEG 2000 [12] employs a complex entropy coder, which means that its hardware and software resource requirements make it unattractive for real-time implementations at UHD resolutions. Latency-wise, the JPEG 2000 versatility makes it possible to configure it in a lower latency mode, such as the VSF TR-01 [13] configuration (leading to an end-to-end latency around 256 lines), or even in a mode bringing the latency around 32-line (as explored in Section VIII-A). In this latter case, JPEG XS and JPEG 2000 reach similar quality levels (with JPEG 2000 slightly outperforming JPEG XS), still at the cost of much greater complexity for JPEG 2000. Recently, a lower complexity version of JPEG 2000 has been standardized, namely High-Throughput JPEG 2000 (HTJ2K) [14]. As explained in Section V-G, it lowers the cost of the entropy stage significantly, though at the price of not being able to predict the rate precisely anymore, which induces additional complexity if online CBR coding is targeted. HTJ2K is compared to several other codecs including JPEG XS (although in a non-optimized version) in [15].

[1]The Alliance for IP Media Solutions (AIMS), https://aimsalliance.org/

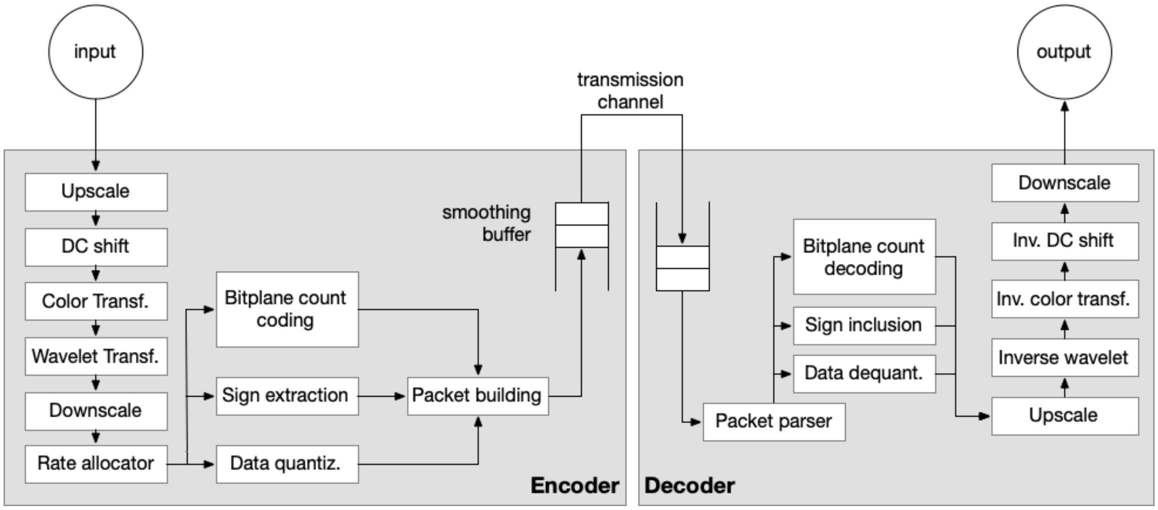


Fig. 1. JPEG XS block diagram for encoder and decoder

The encoder complexity of HEVC [16] is an order of magnitude greater than that of JPEG 2000. While this is acceptable for a distribution (one-to-many) codec and ASIC implementations, it makes it unsuitable as a mezzanine codec. Furthermore, as experiments carried out by WG1 members showed and as reproduced in Section VIII-A, it is not robust to multi-generation compression.

VC-2 [17] is of low complexity, but image quality is considered insufficient. ProRes and JPEG XS give similar results in terms of quality. However, based on the SMPTE disclosure document [18], it is based on macro blocks of 16x16 pixels. This, in combination with the lack of any sophisticated rate control mechanism, defeats latency bound coding. Moreover, the symbol-per-symbol entropy coding makes fast CPU implementations challenging as it offers no inherent parallelism. Other codecs, like DSC [19] and VDC-M [20] do not fit either because they exclusively target ASIC-based display compression, making efficient and cost-effective implementations on FPGAs and GPUs hard to achieve. In addition, they are not optimized for fine-grain parallel processing making fast implementation on multi-core CPU or GPU platforms difficult.

Results of experiments comparing JPEG XS with other codecs are provided in Section VIII.

V. Technical overview

The JPEG XS Core Coding System [21] is a classical wavelet-based still image codec. Even though its typical deployment is in video-over-IP solutions, it does not include a frame-buffer to avoid external memory for a frame buffer for FPGA implementations, making it very cost efficient. A JPEG XS encoder-decoder workflow is detailed below[2] and is presented as a block diagram in Figure 1.

A. Inter- and intra-component decorrelation

At the very first stage, the image data is up-scaled to 20 bits precision, followed by removal of the DC offset, creating a signal symmetric around zero. For RGB input, a reversible color decorrelation transformation into an approximate YCbCr space is run subsequently; this transformation is identical to the Reversible Color Transform (RCT) of JPEG 2000. This is followed by a wavelet transformation; currently, the specification allows a one or two-level vertical transformation with the LeGall 5/3 wavelet also known from JPEG 2000, and up to 8 horizontal decomposition levels. This somewhat asymmetric transformation ensures that the input-output latency does not exceed the maximum allowed end-to-end latency of 32 screen lines. In fact, computations show that the algorithmic latency of the encoder-decoder latency due to the wavelet transformation alone is 9 lines for the two-stage and 3 lines for the one-stage wavelet, leaving the remaining lines for any form of rate allocation that is not further specified in the standard.

The wavelet stage is followed by a pre-quantizer that first represents data as sign and magnitude and then chops off the 8 least significant bit-planes. This can easily be seen to be equivalent to a dead-zone quantizer with a quantization bucket size of $\Delta = 256$ and a dead-zone whose size is twice that of a regular bucket. This pre-quantization is, however, not used for rate-control purposes and only ensures that the following data path can be 16 bits wide, simplifying the requirements on hardware and software implementations.

[2]Subsections V-A to V-C are taken from [22] and included here with permission of the original authors, who are all among the authors of the present paper.

B. Quantization

The actual quantization for rate-control purposes comes next; what is special here is that JPEG XS not only offers a dead-zone quantizer similar to JPEG 2000 by chopping off the T least significant bits, but also a data-dependent uniform quantizer. For that, the pre-quantized wavelet coefficient values are segmented into groups of four coefficients each, the so-called "coding groups". Each coding group defines the quantity M_g as

$$M_g = \max\Big(\lfloor \log_2 \max_{i \in g} x_i \rfloor + 1, 0\Big) \tag{1}$$

where g denotes the coding group and x_i the ith coefficient within g. M_g can be interpreted as the number of populated (non-zero) bit-planes of the coding group and is therefore called the "bit-plane count" within the standard. When the dead-zone quantizer is selected, it divides the full interval $(-2^{M_g}, 2^{M_g})$ of values possible within the coding group by the quantization step size $\Delta = 2^T$, leaving $2^{M_g+1-T} - 1$ buckets. The subtraction of -1 is because the zero bucket has twice the size of a regular bucket and extends from -2^T to 2^T.

Alternatively, when the uniform quantizer is selected, it needs to have the same output range and the same number of buckets as the deadzone quantizer, and hence has to have a bucket size of

$$\Delta_T = \frac{2^{M_g+1}}{2^{M_g+1-T} - 1} = 2^T \left(\frac{2^{M_g+1-T}}{2^{M_g+1-T} - 1} \right) . \tag{2}$$

While quantization is still straightforward to implement by using only shifts and adds, inverse quantization (i.e. reconstruction) requires a trick, namely the Neumann series expansion to avoid any division. This series is explicitly spelled out in the standard. Details are found in [22].

The quantizer is controlled by the rate allocator whose aim is to compress the entire image down to an externally given target bit-rate; in many JPEG XS use cases, this target rate must be met precisely, and it must be met in such a way that the overall end-to-end latency remains limited to the 32 lines quoted above. For that, JPEG XS segments the image similar to JPEG 2000 in the wavelet domain into rectangular regions called "precincts". While in JPEG 2000 precincts are typically quadratic regions in the wavelet domain, a precinct in JPEG XS includes only one or two lines of wavelet coefficients for each band. This can be compared roughly to thin rectangular precincts in JPEG 2000 where the dimension is scaled by 2^r, where r is the resolution level. Rate allocation assigns now to each band b in each precinct p a quantization parameter $T_{b,p}$, except that the rate allocator of JPEG XS is — due to latency constraints — unable to operate on a full frame, but has access only to a limited window of the image. Hence, it is a heuristic, rather than a precise, algorithm.

Now, instead of transmitting the full sets of $T_{b,p}$ separately for each precinct, JPEG XS uses a more compact description and instead only signals two parameters per precinct, denoted as quantization Q_p and refinement R_p, and two additional parameters G_b and P_b per band. The latter are properties of the wavelet filters and thus remain constant throughout the frame. The standard does not say how the G_b and P_b tables are derived, but only describes the procedure as to how the $T_{p,b}$ are derived from them, namely set

$$T_{p,b} = Q_p - G_b - r \tag{3}$$

where r is 1 for $P_b < R_p$ and 0 otherwise. The G_b are called *band gains* and R_p are denoted *band priorities*. They depend only on the wavelet filter and thus only require signaling once per frame.

Hence the quantization parameter $T_{p,b}$ is derived from three quantities: First, a base quantization Q_p that is modulated spatially to ensure a constant output bit-rate; second, a band-dependent gain offset G_b that models the impact of quantization errors in the band b on the overall quality of the reconstructed image; and third, a band-dependent "refinement" r. To motivate r, note that $T_{p,b}$ describes how many bit-planes are truncated away by quantization. If $r = 1$, one additional bit-plane is included for the band b, i.e. the band is "refined". The order in which refinements occur is thus given by P_b, with R_p enumerating how many bands are refined. Thus, if any excess rate would remain available after quantization by truncating $Q_p - G_b$ least significant bit-planes away, R_b bands in the order of P_b are selected to include one additional bit-plane.

This algorithm can be motivated as follows: The ℓ^2 quantization error contribution of wavelet band b in the image domain is approximately given by the ℓ^2 norm of the inverse impulse response γ_b, i.e.

$$\gamma_b := \|W_b^{-1} \delta\|_2 \tag{4}$$

where W_b^{-1} is the inverse (reconstruction) wavelet filter of band b, $\|.\|$ is the square root of the ℓ^2-norm, and δ is a delta-peak (impulse) signal [23] of unit magnitude. This approximation holds exactly if the wavelet coefficients are statistically decorrelated.

Now, for a fixed rate quantizer, the quantization of band b should be derived from a base quantization Δ_0 through

$$\Delta_b = \frac{\Delta_0}{\gamma_b} . \tag{5}$$

Even though JPEG XS uses an (albeit simple) entropy coding, and hence cannot be precisely described as a fixed-rate quantizer, this still holds as an approximation for high bit-rates, see for example [12], [23].

Unfortunately, a fully-fledged rate-distortion optimization, i.e. ensuring MSE optimality under a rate-constraint, is far too complex for JPEG XS purposes and so JPEG XS only attempts to reach the target rate by modulating the band quantization parameters $T_{p,b}$ without actually measuring the distortion. Now recall that the quantizer bucket size $\Delta_{p,b}$ is proportional to $2^{T_{p,b}}$, and thus any power of 2 in γ_b can be factored out into the truncation parameter selection $T^0_{p,b}$, i.e.

$$T^0_{p,b} \approx \lceil \log_2 \Delta_{p,b} \rceil = \left\lceil \log_2 \frac{\Delta_p}{\gamma_b} \right\rceil \approx \lceil \log_2 \Delta_p \rceil - \lceil \log_2 \gamma_b \rceil = Q_b - G_b \tag{6}$$

where Δ_p is the precinct (but not band)-dependent base-quantization into which any proportionality factors are absorbed, and γ_b is the ℓ^2 norm of the inverse impulse response. Thus, clearly, $G_b = \lceil \log_2 \gamma_b \rceil$.

As in JPEG 2000, quantization selection through Q_b and thus through the bit-plane alone would, however, be too coarse and is not sufficient to reach the desired target rate precisely enough. Since the γ_b are not exact powers of 2, we have not yet made use of the remainders after splitting off the leading power of 2 of γ_b into G_b. Thus, define

$$\mu_b := \gamma_b / 2^{G_b} \ . \tag{7}$$

The JPEG XS standard does not signal μ_b directly. Instead, note that eq. (5) implies that quantization of bands with a smaller μ_b provides better image quality and should be refined first if excess rate is available.

It is thus sufficient to sort bands in order of μ_b, and assign them priorities P_b according to their relative rank on this sorted list. If for a given set of $T^0_{p,b} = Q_p - G_b$ additional rate remains available, the excess rate can be made use of by including one additional bit-plane in the order given by μ_b.

The precise manner in how a rate allocator operates is not specified in the standard and multiple algorithms can be considered. In this work, the input image is coded top to bottom, precinct by precinct, and rate allocation proceeds in a window containing multiple precincts sliding over the wavelet-transformed image data. As new data arrives, it is added to the bottom of the window. The topmost precinct of the window is encoded and removed from the window. The window size is selected to be just sufficiently small that, including the latency of the wavelet analysis and synthesis filters, the maximum end-to-end latency is still bounded by 32 lines.

Then, for each window position, (Q_p, R_p) are selected such that the bandwidth limit for this window is not exceeded, where the available bandwidth is determined by the bandwidth of a constant bit-rate transmission channel and the available capacity in the output smoothing buffer of the encoder. The size of this buffer matches that of the decoder smoothing buffer, as specified in the JPEG XS standard, part 2; see section V-F for more details. The channel capacity is given by the target bit-rate of the image and the image size.

This strategy is relatively simple, and it does not include a frame buffer; this is ideal for low-cost FPGA solutions where external memory should be avoided, and the entire rate allocation window can remain cached in local block-RAM on the chip. Other strategies, including using a heuristic from past frames or previous lines, are of course also possible.

C. Entropy coding

Rate allocation is followed by entropy coding. Due to the relative simplicity of the entropy coding mechanism described below, the rate allocator does not need to go through this step as it would in JPEG 2000. Instead, it can compute precisely for any operating point (Q_p, R_p) of the quantizer the output rate without performing the actual encoding step. Some insights into how the rate computation can be performed quickly will be given in section V-E.

The entire encoding process is designed to be pipelined in typical applications, and thus proceeds over the data line by line. As a line is output by the rate allocator, it is quantized and quadruplets of wavelet coefficients are formed. These are the same groups that have already been set up for parametrizing the data-dependent quantizer. For each group, the following data sets are formed: First, the bit-plane counts of all groups; second, the quantized coefficient values themselves, and third the signs of all non-zero coefficients in the group. Of all three data sets, only the first is entropy coded, and all remaining ones are transmitted directly, without any coding. Figure 2 shows a precinct resulting from a 5h1v decomposition and a coding group within.

Though it seems slightly counter-intuitive, the first group — the bit-plane counts — requires a major part of the overall rate. Due to the energy compaction property of the wavelet, typically no more than two bit-planes have to be included at all. The standard allows the bit-plane count of the current line to be predicted from the line above, coding only the prediction residual, and in addition, to code runs of zero-predicted or zero bit-plane counts by a single bit.

Prediction, if enabled, works as follows. Given the bit-plane count of the current group M_g, the bit-plane count of the group above $M_{g'}$ and the truncation positions $T_{p,b}$ of the current and the past precinct $T_{p',b}$, then the prediction residual $\delta_{g,g'}$ is defined as

$$\delta_{g,g'} = \max(M_g, T_{p,b}) - \max\bigl(\max(M_{g'}, T_{p',b}), T_{p,b}\bigr) \tag{8}$$

This looks curious and requires some explanation: first, given all truncation positions and $M_{g'}$, the formula mapping M_g to $\delta_{g,g'}$ is clearly invertible as long as $M_g \geq T_{p,b}$. If $M_g < T_{p,g}$, all data has been clipped away and the coding group has been quantized to zero, so the decoder does not need any more precise a value of M_g to operate. The maximum over $M_{g'}$ and $T_{p',b}$

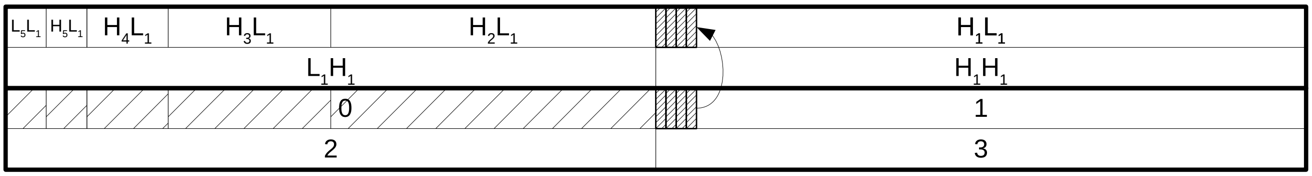


Fig. 2. Two adjacent precincts for a 5h1v decomposition, with one coding group in the H_1L_1 band. Each precinct is 2 lines high, i.e. each box contains coefficients of one line. bit-plane counts can be predicted vertically as shown, except for the first precinct of a slice. The bands comprising the low-pass packet of the second precinct required for proxy extraction is shaded; these form the first packet. The top precinct shows the band decomposition, the bottom precinct the packet indices.

ensures that it does not even require any more precise a value for inverse prediction. Finally, also including the maximum over $T_{p,b}$ in the predictor ensures that the prediction becomes zero if the quantization is large enough and is otherwise monotonically decreasing in $T_{p,b}$, which is important for rate control purposes.

The prediction residual is now encoded by a unary code that consists of two sub-alphabets. To this end, note that the residual is bounded from below by

$$\delta_{g,g'} \geq \Theta_{g',p',p,b} := \min\bigl(T_{p,b} - \max(M_{g'}, T_{p',b}), 0\bigr) \ . \tag{9}$$

This is because, if $T_{p,b} \geq \max(M_{g'}, T_{p',b})$, then we have

$$\delta_{g,g'} = \max(M_g, T_{p,b}) - T_{p,b} \geq 0 \tag{10}$$

and if $T_b < \max(M_{g'}, T_{p',b})$, then

$$\delta_{g,g'} = \max(M_g, T_{p,b}) - \max(M_{g'}, T_{p',b}) \geq T_{p,b} - \max(M_{g'}, T_{p',b}) \tag{11}$$

, which establishes the claimed relation.

The encoding now works as follows. For

$$|\delta_{g,g'}| \leq |\Theta_{g',p',p,b}| = \max\bigl(\max(M_{g'}, T_{p',b}) - T_b, 0\bigr) \tag{12}$$

both positive and negative values for $\delta_{g,g'}$ are possible, for which a Golomb-Rice code of the form

$$\begin{aligned} 1^{2|\delta_{g,g'}|-1}0 \quad &\text{for} \quad \delta_{g,g'} < 0 \quad \text{and} \\ 1^{2|\delta_{g,g'}|}0 \quad &\text{for} \quad \delta_{g,g'} \geq 0 \quad \text{and} \quad |\delta_{g,g'}| \leq |\Theta_{g',p',p,b}| \end{aligned} \tag{13}$$

is used. In other words, code-words are consecutively assigned to an interleaved sequence of negative and positive code-words with negative code-words getting the shorter code-word, and the zero prediction encoded by a single zero bit.

If $|\delta_{g,g'}| > |\Theta_{g',b,b}|$, negative code-words are no longer possible and a single-sided code is sufficient. The encoding is then

$$1^{\delta_{g,g'}+|\Theta_{g',p',b}|}0 \quad \text{for} \quad |\delta_{g,g'}| > |\Theta_{g',p',p,b}| \tag{14}$$

This code is clearly decodable and has much in common with the Golomb-Rice coding of JPEG-LS [24] except that it does not waste coding space for large positive residuals.

To further improve compression efficiency, the encoder forms *significance groups* out of 8 consecutive coding groups. Depending on an option signaled in the frame header, a significance group is called *insignificant* if either all bit-plane counts of its contributing coding groups vanish, or all prediction residuals vanish. If significance coding is used, an upfront pass over the data transmits the significance information, one bit per significance group. bit-plane count information for insignificant groups is then not transmitted in the follow-up bit-plane count coding pass.

In total, the rate allocator can select between four regular prediction modes per wavelet band — with or without prediction, and with or without significance coding — and one of two significance coding methods per frame, namely whether zero predictions or zero counts are coded. One additional flag also allows any bit-plane count coding at all to be disabled, spending a fixed rate of four bits per bit-plane count. This "fallback mode", also called the *raw mode*, ensures that a decoder only needs a fixed-size buffer for entropy decoding; the rate allocator must select this mode as soon as the regular coding modes overrun the rate of the raw-coded pass.

D. Packetization

The JPEG XS codestream syntax follows the guidelines of JPEG Systems: It is a typical *marker-based* bitstream whose syntactical elements are introduced by a unique two-byte indicator starting with a 0xff byte and a second byte indicating the type of *marker*, followed by a two-byte length field. This allows high-level syntax parsing without requiring a full decoding step. Unlike other standards, e.g. JPEG or JPEG 2000, JPEG XS does not attempt to avoid spurious marker sequences in entropy-coded data, i.e. there is no byte-stuffing or bit-stuffing procedure that prevents the entropy coder from generating byte sequences that re-assemble markers. This decision was made to keep the rate allocator as simple as possible since predicting the necessity for byte- or bit-stuffing would be relatively expensive. Note again that due to the simplicity of the encoding, the rate-allocator does not need to go through the actual encoding phase, see section V-E for details.

The encoded image data is grouped into slices, each 16 image lines high. Except for the wavelet transformation, which runs across slices, slices can be decoded independently and thus offer a mechanism for parallelizing encoder and decoder; in particular, the vertical prediction mode of bit-plane count coding has to be disabled at the start of each slice.

Each slice consists of multiple precincts, where a precinct consists of all wavelet filter outputs; a precinct encodes all wavelet coefficients that contribute to a long horizontal stripe of the image. The concept of a precinct is comparable to that of JPEG 2000 except that JPEG XS precincts always extend over the full image width, but cover only 2^v lines, where v is the number of vertical decomposition levels, see Figure 2. The quantization parameters Q_p and R_p are signaled within the precinct header, along with the prediction modes for the bit-plane count coding.

Precincts consist of a precinct header indicating its size and multiple *packets*. The precinct header enables implementations to skip quickly over precincts, and hence to start decoding at a particular slice, allowing efficient multi-threaded implementations on CPUs. The precinct size is an output, or a by-product of the rate allocator which needs to compute all rates for precise rate control purposes; size computation does not therefore increase latencies beyond the latency imposed by the rate allocator.

A packet contains the entropy coded data from a single line, and one or multiple bands of the precincts. The purpose of packets is to allow proxy extraction, i.e. to easily access a low resolution image from a high resolution bitstream. For example, for a wavelet decomposition using one vertical level, JPEG XS uses four packets: the low-resolution image in the first packet, and three higher-resolution bands that include all necessary wavelet coefficients to allow the reconstruction of a full-resolution image from the lower-resolution first packet. That is, the additional three packets cover only one wavelet band per component, whereas the first low-resolution packet may contain more than one band, depending on the number of horizontal decompositions. Thus, each packet covers approximately the same number of wavelet coefficients, but — due to the energy compaction property of the wavelet — will not have equal size. Most data is spent by the first packet.

Packets consist of a packet header and multiple *subpackets*, where the packet header encodes the sizes of all its subpackets. This information is also a by-product of the rate allocation process and thus readily available during encoding. Each subpacket contains the data from one of the coding passes: An optional significance subpacket, which contains the significance flags indicating whether coding groups have zero bit-plane count or zero bit-plane count prediction residuals, the bit-plane count packet, the data packet, and an optional sign packet including the signs of all non-zero wavelet coefficients in the data subpacket. Whether a sign packet is included or not is also signaled once per frame; if it is not, signs are embedded in the data packet, and transmitted last. To avoid a closed feedback loop in the FPGA design which would limit the maximum attainable operating frequency, sign bits are in this case always written, regardless of whether coefficients are quantized to zero or not. If a sign packet is written, only the signs of non-zero coefficients are transmitted.

E. Rate computation

As stated in section V-D, the rate allocation process needs to compute precisely the exact rate of each subpacket of a precinct, not only to select the quantization parameters (Q_p, R_p), but also to be able to create the subpacket size information required by the packetization step. This section will provide some insight into this mechanism.

First, note that the sizes of the packet, precinct and slice overhead are constant and thus readily available. The size of the significance subpacket depends only on the number of coefficients in the packet — namely one bit for each group of 8 coding groups — and is not even signaled since the decoder can also infer it from the geometry of the wavelet bands.

The computation of the rate of the remaining subpackets, namely the bit-plane count subpacket, the data subpacket and the sign subpacket, seems to require an algorithm of the order $O(N \times W)$ where N is the number of operating points of the quantizer, i.e. the number of choices for (Q_p, R_p), and W is the number of coefficients. That is, it seems that rate allocator has to go through a provisional encoding step for each possible choice of quantization parameters to determine the length of the subpackets. Interestingly, this is not the case, and the complexity of the rate allocation can be reduced to $O(N+W)$ where a first preparatory step collects statistical information on the data to be encoded — this is an $O(W)$ step — and then actual rate allocation just selects (Q_p, R_p) values and examines a look-up table to find the corresponding rates. This is an algorithm of complexity $O(N)$.

How this preparation step looks shall now be demonstrated on the data subpacket; similar considerations can be made for all other subpackets. For this, denote by x_i, $i \in (0, W-1)$ the incoming (pre-quantized) wavelet coefficients entering the rate allocation phase. The number of bit-planes a code group j occupies is then given by:

$$B(0,j) := \left\lfloor \max\left(\log_2\left(\max_{i=0}^{3} x_{i+4j}\right) + 1, 0\right)\right\rfloor \,, \tag{15}$$

where we use the convention $\max(-\infty + 1, 0) = 0$ in the event that all coefficients are zero.

For the time being, let us restrict ourselves to deadzone quantization, which truncates the least significant $T(Q_p, R_p)$ bits of all coefficients. Thus, the number of bit-planes that remain available after quantization is

$$B(T,j) := \max\left(\left\lfloor \max\left(\log_2\left(\max_{i=0}^{3} x_{i+4j}\right) + 1, 0\right)\right\rfloor - T, 0\right) = \max\bigl(B(0,j) - T, 0\bigr) \tag{16}$$

Hence the number of bits the data subpacket occupies without sign information is given by

$$R_D(T) = 4 \sum_{j=0}^{W-1} B(T,j) \ , \tag{17}$$

where W is the number of code groups in the data subpacket whose size is to be determined. For embedded sign coding, the total data subpacket rate is then $R_D(T) + 4W$ since 4 additional bits per code group are spent to encode the signs. We will see below how to compute the rate of the sign subpacket for separate sign coding.

The sum in eq. (17) can now be computed in a different order by first summing over all code groups that contribute b bit-planes, then summing over the necessary b. To this end, denote by χ_f the characteristic function which is 1 if the expression f is true, and 0 otherwise. The number of code groups n_b having b bit-planes can then be written as:

$$n_b = \sum_{j=0}^{W-1} \chi_{B(0,j)=b} \tag{18}$$

where $B(0,j)$ as in eq. (15), and thus the rate $R_D(T)$ can then be written as

$$R_D(T) = 4 \sum_{b>T} (n_b - T) = 4 \sum_{b>T} \left(\sum_{j=0}^{W-1} \chi_{B(0,j)=b} - T \right) \ . \tag{19}$$

The following important observation can be made: the inner sum over j no longer depends on T; rather, it describes the statistics of the coefficients and can be pre-computed before rate-allocation starts. Its computation is of complexity $O(W)$. The outer sum over bit-planes b no longer depends on W, and has to be evaluated once per operating point visited by the rate allocator; it is thus of complexity $O(N)$. Hence, rate allocation for the deadzone quantized data subpacket is of complexity $O(N + W)$, as claimed.

Now, for uniform quantization, the reader may recall eq. (2), namely that the uniform quantizer is data-dependent. In particular, the number of buckets of the uniform quantizer is exactly identical to that of a deadzone quantizer of the same bit-plane count, and thus the data rate of the data subpacket generated by the uniform quantizer is *identical* to that of the deadzone quantizer from which it is derived. In fact, this is the motivation for the design of the uniform quantizer.

The size of the sign subpacket for separate sign coding is also easily derived using a similar trick by first collecting the statistics on the data and rewriting the sums. The important observation here is that only those coefficients contribute to the budget of the sign subpacket by one bit that are not in the deadzone of the corresponding quantizer. Thus, the rate allocation process first proceeds by forming classes of coefficient magnitudes that are quantized to zero for a particular choice of $T(Q_p, R_p)$, i.e. coefficient magnitudes smaller than $\Delta_T = 2^T$ for the deadzone quantizer, and coefficient magnitudes smaller than

$$\frac{\Delta_T}{2} = \frac{2^{M_g}}{2^{M_g+1-T} - 1} = 2^T \left(\frac{2^{M_g-T}}{2^{M_g+1-T} - 1} \right) \tag{20}$$

for the uniform quantizer, see eq. (2). This preparatory step is again an algorithm of $O(W)$ complexity. To compute the number of sign bits written for a particular choice of T, only the number of coefficients in class T needs to be known. As there are at most N possible choices for T, the actual rate allocation step once the classes are prepared is then again only of complexity $O(N)$. Unlike in the data subpacket case, the classes to be formed here depend on the choice of the quantizer.

Finally, for the bit-plane count subpacket, similar classes can be formed, where each class is determined by the length of the VLC code given in section V-C. The computations for this case are similar but lengthy, and are omitted for the sake of brevity.

F. Smoothing buffer and decoding steps

Once packets have been assembled to form the final codestream, it can be sent through the transmission channel. A smoothing buffer ensures a constant bit rate at the output of the encoder, although the input image might consist of input regions that are easier to compress, and others that require more bits per pixel. A separate part of the standard (JPEG XS Part-2 ”Profiles and buffer models” [25]) specifies exactly how large the smoothing buffer is, and in which units data enters and leaves it; this also has implications on the rate control algorithm. JPEG XS is the first codec developed by the JPEG committee that specifies buffer control in such detail as to allow implementations with limited, but well-controlled resources.

Given that the decoder should be able to process the pixels with a constant clock frequency, the number of bits read per time unit varies depending on whether a current wavelet coefficient is easy to compress or not. These rate variations are again compensated by a smoothing buffer at the input of the decoder. A packet parser splits the bit stream into individual data chunks representing parts of a sub-band before the wavelet coefficients are decoded, inversely quantized and transformed back into the spatial pixel domain.

G. System integration and encoder architecture

As seen in section V-E, rate computation can be split into two stages: A statistics collection step that can be merged with the output of the last wavelet filter, and a low-complexity rate allocation step that requires only a table lookup into the statistics collected earlier; this step computes the rate for a given quantizer operating point by means of a table lookup. JPEG XS rate allocation is therefore almost immediate and encoding can proceed without much delay after having filtered all coefficients of a precinct. At the same time, its rate computation is bit-precise as required by an on-line constant-bitrate codec; hence, a JPEG XS encoder can always ensure that a decoder input buffer will never underflow or overflow.

JPEG 2000 provides such a bit-precise rate control as well, however at the cost of a more complex sub-bitplane entropy coding algorithm. Complexity is here generated *before* rate allocation. High-throughput JPEG 2000 lowers the cost of the entropy coding stage significantly, though at the price of not being able to predict the rate precisely anymore, and either requiring a heuristic for rate estimation that might fail in corner cases, or requiring several iterations of its combined quantization and entropy coding stage. Hence, if precise rate control is required as for online CBR coding, quantization and entropy coding must be potentially repeated over multiple iterations. Therefore, additional complexity here generated *within* rate allocation. JPEG XS is unique in the sense that due to the simplicity of its entropy coding stage, precise rate control becomes trivial without requiring additional complexity upfront, though at the cost of reduced coding efficiency.

Additional constraints arise in real-time FPGA or hardware implementations: Here, rate control *must* complete within a limited time window to enable constant throughput of image data, and multiple iterations over encoding and rate control may not be possible.

VI. Latency analysis

As stated in the introduction, JPEG XS is designed to compress its input signal with a latency of a few lines at most. In this section, we analyze in how far JPEG XS is able to meet its design goals of an end-to-end latency of 32 lines.

For this analysis, a couple of assumptions are made: first of all, we do not take the latency of the transmission channel itself into account during the analysis as it is beyond the control of the codec. Furthermore, the transmission channel bandwidth is considered to be at least equal to the encoded bit-rate. Under these assumptions, the sources of latency are as given in table II: the forward and inverse color transformations do not require pixels to be buffered and thus only cause a latency of a few pixels. The forward and inverse vertical decomposition of the wavelet filter requires 2 to 6 lines with the 5/3 filter, depending on the number of decomposition levels. With vertical decomposition disabled, only the latency of the horizontal filter needs to be included, causing a latency of less than a single line. Rate allocation requires at least a single precinct to be buffered in order to be able to compute the rates of all contributing packets. However, in practice, an implementation typically buffers as many precincts as possible within the target latency bound, i.e. it would make the look-ahead window of the rate allocation step as large as permissible. Following this step, or even before rate allocation, data needs to be reordered from the output order of the wavelet into the packet order required for transport. This accounts for an extra single precinct latency. Similarly, inverse data reordering at the decoder side also contributes by a single precinct. Quantization and entropy coding only require a few pixels of latency, keeping in mind that the rates of the packets have been previously computed by the rate allocation process. The same holds for entropy decoding and inverse quantization. Finally, the amount of buffering between the encoder and the decoder combined, i.e. the size of the smoothing buffer, accounts for 1 precinct and is defined in JPEG XS Part-2 [25].

As seen in table II, the lower bound of the end-to-end latency is between 5 columns and 29 lines, depending on the number of vertical wavelet decompositions, and thus the height of a precinct in lines. Realistically, such low latencies can only be obtained in hardware implementations; in software implementations, the latency can be as low as this, but then the codec is restricted to operate in a single thread with only data parallelism available — i.e. SIMD instructions — to speed up processing. Throughput in such implementations is often too low for real-time processing of frame sizes larger than full HD (1920×1080 pixels), even on modern processors. To speed up processing further, work needs to be offloaded to multiple CPU cores.

In JPEG XS, the smallest independently codeable unit is a slice, and thus each core typically operates on a subset of the slices of the source image. Due to the length of the vertical wavelet filter, the image domains on which the cores operate has to overlap by a couple of lines, i.e. some image lines are transformed twice by two distinct cores. To avoid synchronization overheads, rate allocation is also performed by each core separately, where each core gets its own rate pool. Consequentially, rate will not be moved from one thread to another, which results in a slight degradation in quality.

Such an implementation was studied in [26], and an end-to-end latency between 70 and 350 lines, depending on the configuration, was reported. The latency also depends on the transport mechanism and whether all packets need to be transported sequentially, or if out-of-order transport is permissible. In fact, the corresponding IETF draft for JPEG XS transport over RTP allows both transport modes [27]. The overall drop in quality due to separate rate allocation per core was found to be below 0.1dB. For details, we refer the reader to the reference.

For GPU implementations, typically entire frames are up- and downloaded to the GPU since starting compute kernels on the GPU is relatively slow; the overhead to initiate a computation is typically too high to allow sub-frame latency. Furthermore, the data upload and download is limited by the (relatively) slow bus between GPU and the host system; while the time for data transfer can be hidden by processing another frame on the GPU in parallel with the transfer, this frame adds to the total latency. GPU implementations therefore typically have a latency of multiple frames.

TABLE II
SOURCES OF LATENCY FOR 0, 1 AND 2 VERTICAL DECOMPOSITION LEVELS. FOR 0 LEVELS, COLUMNS ARE ASSUMED, AND A PRECINCT EXTENDS OVER ONE COLUMN HORIZONTALLY, AND 1 LINE VERTICALLY. OTHERWISE, PRECINCTS EXTEND OVER THE FULL WIDTH OF THE IMAGE AND ARE 2 OR 4 LINES HIGH (FOR 1 OR 2 VERTICAL DECOMPOSITION LEVELS, RESPECTIVELY).

Latency source	Amount of latency
Color transformation	A few pixels
Wavelet transformation	A few pixels for 0 vertical decomposition
	2 lines + a few pixels for 1 vertical decomposition
	6 lines + a few pixels for 2 vertical decompositions
Rate allocation	1 precinct + a few pixels
Data reordering	1 precinct + a few pixels
Quantization	A few pixels
Entropy coding	A few pixels
Smoothing buffer	1 precinct
Entropy decoding	1 column
Inverse quantization	A few pixels
Data reordering	1 precinct + a few pixels
Inverse wavelet transformation	A few pixels for 0 vertical decomposition
	2 lines + a few pixels for 1 vertical decomposition
	6 lines + a few pixels for 2 vertical decompositions
Inverse colour transformation	A few pixels
Total	5 columns + a few pixels for 0 vertical decompositions
	13 lines + a few pixels for 1 vertical decomposition
	29 lines + a few pixels for 2 vertical decompositions

VII. PROFILES, LEVELS, AND FORMATS

Particular application domains may enforce additional constraints on the encoder and decoder, such as even lower complexity, or limitation of buffer or screen sizes. To this end, the second part of the JPEG XS standard [25] introduces profiles, levels and sublevels (in addition to the specification of the buffer model mentioned in Section V-F).

Profiles define restricted tool sets, and hence reduce the complexity of encoder and decoder by only offering a confined set of coding tools. Profiles are structured along the expected number of logic elements, the expected memory footprint, and whether chroma subsampling or an alpha channel is required. As such, the new International Standard spells out eight profiles, whose characteristics and targeted use cases are summarized in Table III. As seen in the table, profiles are structured along the maximum bit precision of components, the quantizer type, the size of the smoothing buffer and the number of vertical decomposition levels. The smoothing buffer size is given in units of compressed data lines at the target compression rate, or compressed data columns in the case of 0 vertical decompositions.

Coding tools such as embedded or separate sign coding, or whether insignificant coding groups are formed by zero coefficients or vanishing prediction residuals make less of a difference in terms of decoder complexity, and are therefore not restricted by the profile. The performance of various entropy coding options was first reported in [22].

Concerning content type, the ”light” set of profiles will perform best on natural content as the limited smoothing buffer size does not allow a rate allocation to be implemented that is sufficiently efficient to deal with the sharp edges usually found in CGI and screen content. The last row of Table III gives an indication of the typical use cases suited to each profile that were used by JPEG experts to structure the profile space.

With profiles selecting the coding features, levels and sublevels on the other hand limit the buffer sizes. More precisely, levels impose restrictions in the uncompressed image domain and sublevels in the compressed domain. Thus, in particular, levels constrain the frame dimensions and the refresh ratio in approximately the same way as HEVC levels (e.g. 1920/60p as maximum input format).

In the third part of the JPEG XS Standard [28] and in other standardization activities, various file formats and transport formats are specified, allowing one or more JPEG XS codestreams to be stored or streamed. These formats are based on already existing format syntax, and usually include additional metadata such as color space information and frame rate, allowing color-correct rendering on computers and archiving or streaming of videos. The currently defined formats are listed in Table IV.

To ease integration of JPEG XS workflows, previous encapsulation work carried out for other codecs has been re-used as much as possible. This is, for instance, the case of the MXF wrapper or the MPEG-2 TS wrapper that are based on what has been done previously for JPEG 2000. All these formats have been defined in a consistent way, allowing seamless re-encapsulation operations when switching from one format to another without the need for a deep codestream inspection.

VIII. PERFORMANCE EVALUATION

This section provides experimental results illustrating the claims being made for JPEG XS. First a rate-distortion comparison against other state-of-the-art compression technologies used in the market is given, as well as an objective evaluation of the multi-generation robustness. This is followed by a short complexity analysis, and finally subjective evaluation results are discussed.

TABLE III
JPEG XS PROFILES AND THEIR CONFIGURATION. IF A PARTICULAR CONFIGURATION ITEM IS NOT EXPLICITLY STATED, ALL POSSIBLE CHOICES ARE AVAILABLE.

Profile	Main 422.10	Main 444.12	Main 4444.12	Light 422.10	Light 444.12	Light-Subline 422.10	High 444.12	High 4444.12
Bit depth	8, 10	8, 10, 12	8, 10, 12	8, 10	8, 10, 12	8, 10	8, 10, 12	8, 10, 12
Chroma sampling formats	4:0:0 4:2:2	4:0:0 4:2:2 4:4:4	4:0:0 4:2:2 4:4:4 4:2:2:4 4:4:4:4	4:0:0 4:2:2	4:0:0 4:2:2 4:4:4	4:0:0 4:2:2	4:0:0 4:2:2 4:4:4	4:0:0 4:2:2 4:4:4 4:2:2:4 4:4:4:4
Color transformation	No	RCT if 4:4:4	RCT if 4:4:4	No	RCT if 4:4:4	No	RCT if 4:4:4	RCT if 4:4:4 or 4:4:4:4
Vertical wavelet decompositions	0-1	0-1	0-1	0-1	0-1	0	0-2	0-2
Quantization type	DZQ or Uniform	DZQ or Uniform	DZQ or Uniform	DZQ	DZQ	DZQ or Uniform	DZQ or Uniform	DZQ or Uniform
Smoothing buffer size	16	16	16	4	4	2	16	16
Target content type	Natural, CGI and screen content			Natural content		Natural content	Natural, CGI and screen content	
Target use cases	Broadcast, Pro-AV, Frame Buffers, Display links			Broadcast, industrial cameras, in-camera compression		Cost-sensitive applications	Same as "Main" but for high-end devices, contribution, cinema remote production	

TABLE IV
JPEG XS TRANSPORT AND CONTAINER FORMATS

Format (+ext. if applicable)	Type	Description	Reference std document
JXS (*.jxs)	JPEG XS file format	Storing of single JPEG XS images	ISO/IEC 21122-3 [28]
MP4 (*.mp4)	ISO Base Media File format (ISOBMFF)	Storing of JPEG XS video	ISO/IEC 21122-3 [28]
HEIF (*.heif)	High Efficiency Image File Format	Storing of mixed image and video content	ISO/IEC 21122-3 [28]
MXF (*.mxf)	Material Exchange format	MXF wrapper for JPEG XS	SMPTE ST 2124
MPEG-2 TS	Transport stream	MPEG-2 Transport stream wrapper for JPEG XS	ISO/IEC 13818-1:2019/AMD1
RTP	Real-time Transport Protocol payload format	IP transport for JPEG XS	IETF RFC (Last Call status, exp: Q2 2021)
ST 2110-22	System stream	Encapsulation of compressed video stream in SMPTE 2110	SMPTE ST 2110-22

A. *Objective assessment*

The objective assessment setup is based on that used for the various JPEG XS Core Experiments performed during the development of the Standard within the JPEG Committee. PSNR comparisons are performed on the small set of still images shown in Figure 3. This set focuses on RGB 4:4:4 8-bit content, as results on the complete set used for the Core Experiments have shown that similar results are obtained for other kinds of content (such as 10-bit or 12-bit, YCbCr, or 422 downsampling). As indicated in Figure 3, this set contains natural, artificial and desktop content.

Figure 4 shows the respective rate-distortion performance of JPEG XS using the Main 444.12 and High 444.12 profiles (both with uniform quantization)[3], and two low-latency JPEG 2000 configurations[4], for bit-rates ranging from 2 to 8 bits per pixel

[3]Created by the JPEG XS reference implementation with an optimized rate-allocation system and using the aforementioned XS profiles. This implementation is available as a free encoder-decoder trial from intoPIX.

[4]Using Kakadu v8.0.5 with command line:

```
kdu_compress -i <ppm> -o <j2c> -rate <rate> -full -precise -no_weights Creversible=no
Corder=PCRL Clevels=5 Cprecincts="{8,8192},{4,8192}" Qstep=0.00001 Kkernels:I2=I5X3
Cdecomp="B(-:-:-),H(-),H(-),H(-),H(-)" Cblk="{4,1024}" Catk=2 Scbr="{1,10}"
```

and:

```
kdu_compress -i <ppm> -o <j2c> -rate <rate> -full -precise -no_weights Creversible=no
Corder=PCRL Clevels=5 Cprecincts="{8,8192},{4,8192},{2,8192}" Qstep=0.00001 Kkernels:I2=I5X3
Cdecomp="B(-:-:-),B(-:-:-),H(-),H(-),H(-)" Cblk="{4,1024}" Catk=2 Scbr="{1,10}"
```

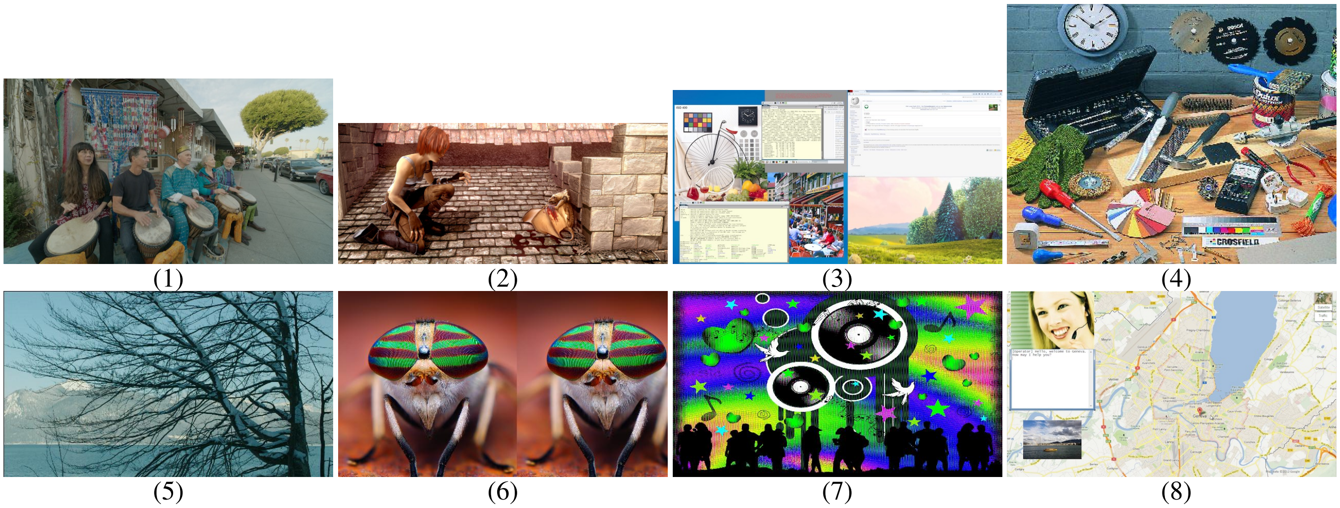


Fig. 3. Still images used for the quality assessments: (1) ARRI_AlexaDrums (3840x2160p, 24, 8b_bt709_444, 00162, natural); (2) BLENDER_Sintel2 (4096x1744p, 24_8b_sRGB_444, 00004606, artificial); (3) RICHTER_ScreenContent2 (4096x2160p, 60, 8b_sRGB_444, 00001, desktop); (4) Tools (1520x1200, 8b, natural); (5) ARRI_Lake2 (2880x1620p, 24_8b_bt709_444, 0040, natural); (6) FemaleStripedHorseFly (1920x1080, 8b, natural); (7) HintergrundMusik (1920x1080, 8b, artificial); (8) HUAWEI_ScMap (1280x720p, 60, 8b_sRGB_444, 000, desktop).

(bpp), which is the usual range for the targeted JPEG XS applications. Both of the JPEG 2000 flavors were carefully designed as a trade-off to keep high quality while achieving end-to-end latencies almost as low as with JPEG XS. These configurations both use thin JPEG 2000 precincts of height 8 and apply a local leaky bucket rate-distortion allocation. The applied wavelet transform in this case is the 5/3 filter with 5 horizontal and 1 or 2 vertical decomposition levels. Note that restricting JPEG 2000 to such a low latency requires using an extension outside JPEG 2000 Part-1 [12], namely an asymmetric number of horizontal and vertical wavelet decompositions (defined in JPEG 2000 Part-2 [29]). The RD curves shown in figure 4 are all very similar, with JPEG 2000 slightly outperforming JPEG XS at the expense of a higher end-to-end latency and much higher complexity (see section VIII-B). In this respect, it should be noted that the end-to-end latency achievable in practice (and indicated in figure 4) in hardware implementations for the two JPEG 2000 configurations is higher than the required 32 lines, being 41 and 51 lines respectively. Finally, it can be seen that having two vertical decompositions with JPEG XS improves the quality at the lower bit-rates. For this reason it is specifically advised that the High 444.12 profile be used in this bit-rate range.

Figure 5 shows three extra rate-distortion results on the Tools image for compression technologies that target the same markets in order to have a more complete comparison. We observe that ProRes[5] reaches the same quality level (at least in the lower bit-rates) as JPEG XS and the low-latency JPEG 2000, but it does not meet the required latency level. The tested VC-2 configuration[6], configured for low-complexity and low-latency, fails to achieve an acceptable quality level.

In addition, Figure 6 shows the PSNR quality over successive generations for JPEG XS, JPEG, JPEG 2000, and a restricted variant of HEVC[7] similar to that which has been proposed as coding technology for JPEG XS. As seen in the figure, while HEVC offers good quality for the first generation, image quality degrades rapidly generation after generation. On the other hand, both JPEG XS and JPEG 2000 have a slight quality drop at the first generation, but further generations remain at an almost constant quality level.

Additional performance results, in particular with regard to the various entropy coding options, can be found in [22].

B. Complexity analysis

As shown, JPEG XS and JPEG 2000 reach similar levels of quality in the range above 2 bpp for a given latency level, but they still involve significantly different levels of complexity. JPEG XS benefits from a very low complexity as every choice in the coding system has been made with the software and hardware implementation constraints in mind. To give a scale of this complexity difference, a comparison has been made between real FPGA core implementations of JPEG XS and JPEG 2000

[5]The applied ProRes implementation did not offer configurable rate-distortion allocation. As such, a script ran all available encoding modes and sorted the resulting outputs by bit-rate. When required, color subsampling (to 4:2:2 or 4:2:0) was performed outside of ProRes. The measured PSNR values represent the end-to-end quality.

[6]Using VC-2 command line options:

```
-f RGB -n 1 -z 8 -d 4 -u 1 -a 1 -k LeGall
```

[7]Using HEVC command line options:

```
-c cfg/encoder_intra_main_scc.cfg --InputChromaFormat=420 --ProgressiveSource
--FrameOnly -cf 420 --FrameRate=30 --FramesToBeEncoded=1 --QuadtreeTULog2MaxSize=5
--GOPSize=1 --IntraPeriod=1 --ConformanceWindowMode=1 --AdaptiveQP=1 --RateControl=0
--TransquantBypassEnable=1 --CrossComponentPrediction=0 --ColourTransform=0
```

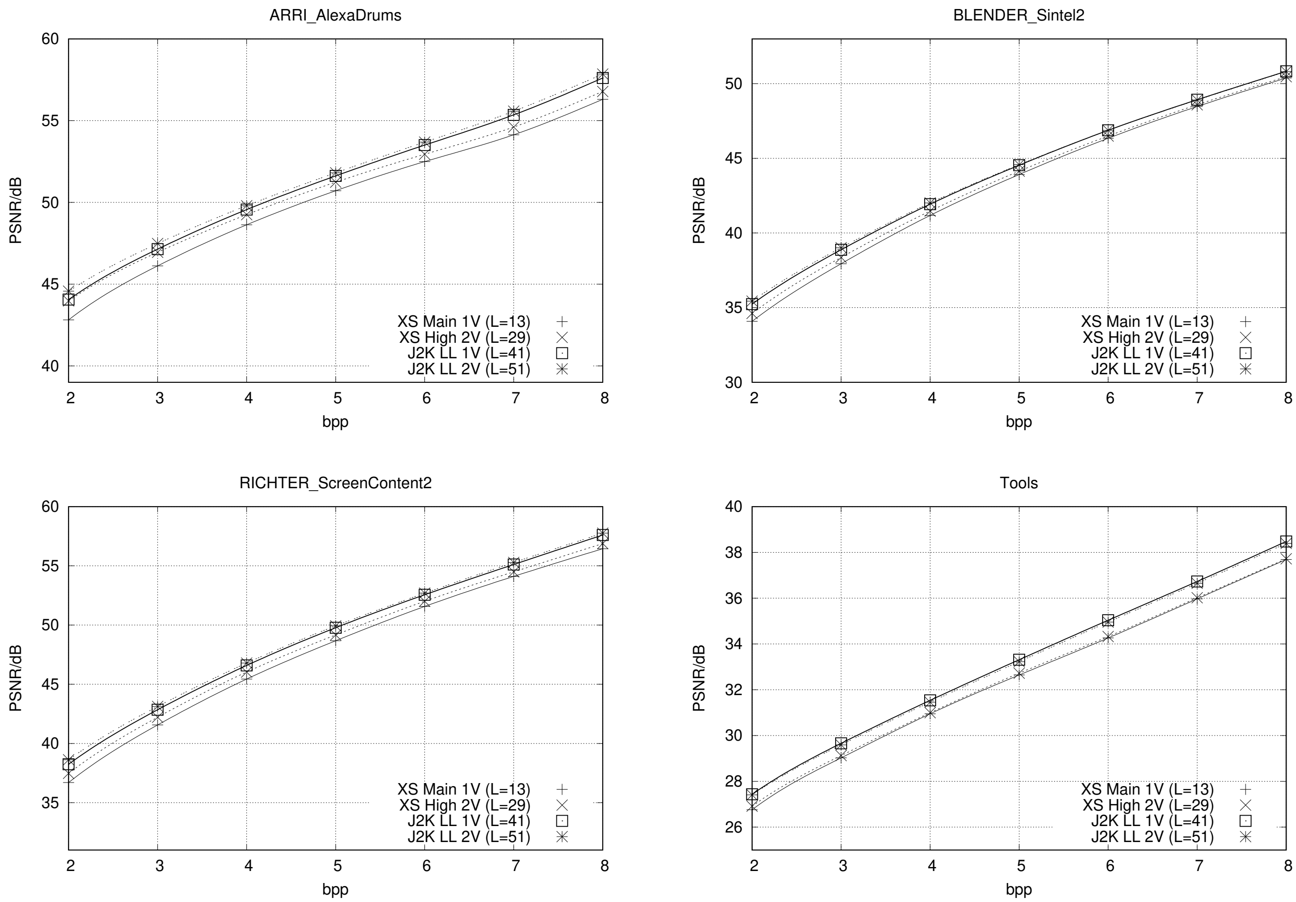


Fig. 4. Rate-distortion performance of various codecs on four images: J2K LL 1V and J2K LL 2V represent JPEG 2000 in low latency modes with respectively one and two vertical decompositions, XS Main 1V is JPEG XS using the main profile (allowing only one vertical decomposition), and XS High 2V is JPEG XS in high profile mode (with two vertical decompositions). Additionally, the configuration labels show the effective implementation end-to-end latency (expressed in lines).

(from the same company). The technical requirements of JPEG XS [1] limit the resources of a 4K 4:4:4 60p encoder or decoder to 25% of an Intel (Altera) Cyclone V 5CEA9, which corresponds to 3,050 Kbits of memory and 56,780 Look-Up Tables (LUTs) [30]. Moreover it does not require any external memory. On the JPEG 2000 side, the same 4K flavour in a general Broadcast profile requires an external memory (DDR), and 95% of an Intel (Altera) Arria V 5AGXA7 [31], which corresponds to 12,977 Kbits and 174,192 LUTs [32]. To take into account the clock frequency difference (100 MHz for the Cyclone, 200 MHz for the Arria), an additional factor of 2 can be inserted to estimate the number of required LUTs for JPEG 2000. Moreover, for JPEG 2000 to reach the multi-generation robustness described in Figure 5, a 32-bit internal precision is enabled (as seen in the Kakadu commandline used for the experiments), which would increase by ∼50% the amount of logic and memory required compared to the default 16-bit precision. In real-life implementations, and depending on the content, sufficient multi-generation robustness can probably be achieved with a bit precision around 20 bits for JPEG 2000. In a configuration leading to a latency similar to JPEG XS, no DDR is required, no DDR controller either (sparing ∼15,000 LUTs @ 200 MHz), but an additional double buffer of 8 4K image lines is needed (∼3,000 Kbits in 16-bit precision). This leads to the typical complexity reduction factors indicated in Table V. Concerning on-chip memory, JPEG 2000 typically requires more memory as the rate and/or code-block size increase: this explains the range of factors (6-12, 4-8). As far as the power consumption is concerned, it increases with the amount of LUTs, the clock frequency, and to a lesser extent with the amount of on-chip memory. The use of external memory also explains the difference between the two JPEG 2000 flavors. As an indication, the DDR for a 4K workflow will typically require more than 4 Watts. Typical consumption of a JPEG XS 4K FPGA encoder is around 0.7 Watt.

C. *Subjective assessment*

As objective metrics cannot accurately predict the perceived quality, a subjective flicker test with the most challenging images, some of them shown in Figure 3, has been performed both at the beginning of the standardization process (with the

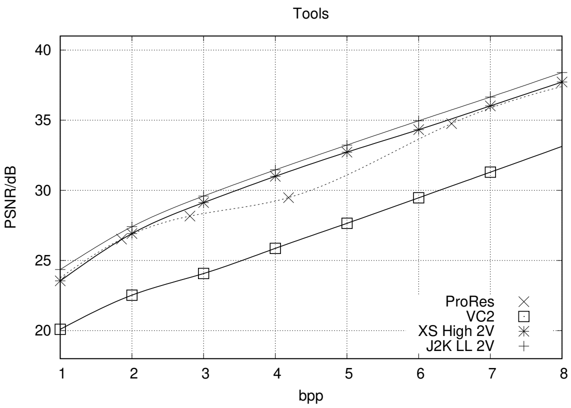


Fig. 5. Rate-distortion performance of various codecs that all target the same markets: J2K LL 2V and XS High 2V are explained in Figure 4. VC-2 [17] and ProRes [18] are also shown, despite the fact that these do not meet the low-latency requirements of JPEG XS.

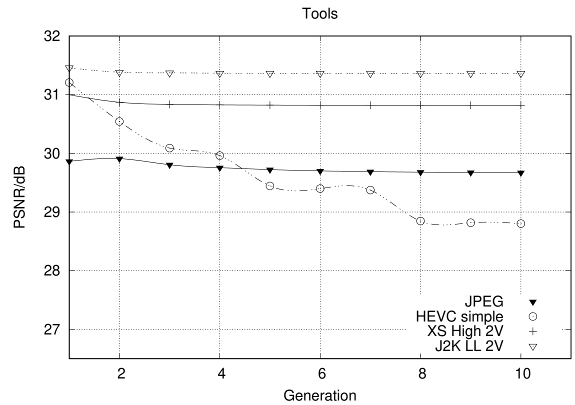


Fig. 6. Generation loss of JPEG XS compared with JPEG, JPEG 2000 (configured for low latency) and HEVC on the Tools image at a bit-rate of 4 bpp.

TABLE V
RESOURCES REDUCTION FACTOR FOR A COMBINED JPEG XS FPGA ENCODER-DECODER COMPARED TO JPEG 2000.

	JPEG XS reduction factor vs low-latency JPEG 2000	JPEG XS reduction factor vs regular JPEG 2000
Logic (LUTs)	÷ 9	÷ 9
On-chip memory	÷ 6-12	÷ 4-8
Power consumption	÷ 6	÷ 10

selected baseline), and at the end (with the improved core coding system eventually standardized). Methodology and results are extensively described in [8] and [33], and a brief summary is presented here.

The test methodology adopted in the JPEG XS subjective assessments follows the recommendations defined in ISO/IEC 29170-2 (AIC Part-2) that provides guidelines for performing the subjective evaluation of near lossless image coding schemes [7]. In particular, the assessment broadly conforms to the test methodology described in Annex B of AIC Part-2. According to this test, the screen is split into two parts. On one side a crop of the original still image is displayed, and on the other side the same crop of the encoded-decoded image is displayed, interleaved with the crop from the original image. Both the encoded-decoded image and its original counterpart appear four times per second on the interleaved side. As a consequence, visual discrepancies between the original image and the encoded-decoded image appear as ”flickering”. The subjects are required to guess on which side of the screen the interleaved content is displayed. Their vote could fall into three categories: (i) *correct*, when interleaved side is correctly identified, (ii) *wrong*, when interleaved side is wrongly identified, (iii) *no decision*, when the subject could not decide which side is interleaved. An empirical metric was then used to obtain a score for each test image:

$$\text{Score} = 2 \times \left(1 - \frac{OK + 0.5 \times ND}{N}\right) \tag{21}$$

where N is the total number of votes for a given stimulus, OK the number of times the flickering is correctly identified and ND the number of times that a subject decided to cast a no decision vote. The score is bounded to $[0, 2]$, where 0 means all subjective votes are correct; score 1 is achieved when all the votes are cast as no decision, implying that no subject could identify flickering (transparent coding). The upper bound score 2 is obtained if all votes are wrong. This case is avoided by using an outlier removal procedure based on the use of control images.

The content selected for this assessment consisted of eight still images, covering a wide range of formats, various content types (i.e. camera-captured, CGI, and screen-content) and presenting several coding challenges (e.g. complex textures or sharp edges). Among them, images (4), (5), (6) and (7) from Figure 3 were used. It must be noted that these images were cropped to help the viewer focus on specific areas of the pictures. The encoded-decoded images were subjected to seven successive encoding-decoding cycles, allowing the *multi-generation robustness* of JPEG XS to be assessed.

The different profiles of JPEG XS (see Table III) were tested and results showed that, except for highly complex images like the FemaleStripedHorseFly image (number (6) in Figure 3), the initial requirement of ensuring full visual transparency up to a compression ratio of 6:1 is met. Successfully passing such a flickering test is of course a very challenging assessment, and not all use cases require that even the highly noisy areas of a compressed image remain identical to the original picture.

This is why, depending on the content used and the target application, higher compression ratios can also safely be used and still ensure a visual quality that is not noticeably different from the original.

IX. STATUS OF THE STANDARDIZATION PROCESS AND UPCOMING EXTENSIONS

Table VI presents the status of all JPEG XS-related standardization activities within the JPEG Committee, as of October 2020. As seen in the table, most parts of the JPEG XS Standard are now either published or in the final stages at ISO (no further technical change is allowed at this stage). Other activities related to transport and container formats occur outside the JPEG Committee, as described in Table IV. Among them, MXF and RTP efforts are currently on-going but they already deal with a mature document and final publication is expected during the course of 2020.

TABLE VI
JPEG XS STANDARDIZATION STATUS AS OF APRIL 2021

Document number	Title	Description	Status
ISO/IEC 21122-1	JPEG XS Part-1: Core coding system	Specifies syntax of JPEG XS bitstream and algoritm for decompression	Published
ISO/IEC 21122-2	JPEG XS Part-2: Profiles and buffer models	Specifies application profiles and buffer model for low-latency	Published
ISO/IEC 21122-3	JPEG XS Part-3: Transport and container formats	Defines file formats and containers for JPEG XS bitstreams	Published
ISO/IEC 21122-4	JPEG XS Part-4: Conformance testing	Specifies method for conformance testing of JPEG XS bitsreams	Published
ISO/IEC 21122-5	JPEG XS Part-5: Reference software	Reference software for JPEG XS	Published
ISO/IEC 21122-1 2nd Edition	JPEG XS Part-1: Core coding system 2nd Edition	Specifies 4:2:0 subsampling and Bayer pattern compression tools	Draft IS[8] (IS exp.: Q2 2021)
ISO/IEC 21122-2 2nd Edition	JPEG XS Part-2: Profiles and buffer models 2nd Edition	Specifies additional profiles for tools introduced in Part-1 2nd Edition	Draft IS (IS exp.: Q3 2021)
ISO/IEC 21122-3 2nd Edition	JPEG XS Part-3: Transport and container formats 2nd Edition	Contains corrigenda and specifies support for 4:2:0 subsampling	Draft IS (IS exp.: Q2 2021)

Within the JPEG Committee, the main activity for JPEG XS is currently focused on the creation of second editions of all parts of the standard. These new editions incorporate all corrigenda that were made to the standard. In addition, they bring support for 4:2:0 chroma subsampling — which was, for historical reasons, absent from the first edition of the Core Coding System —, describe mathematically lossless support (MLS) for up to 12-bit, and provide new coding tools specifically dedicated to compression of Color Filter Array (CFA) data, mostly known as Bayer patterns. In this effort, which started in 2019, several rounds of Core Experiments were organised to identify what needed to be done differently in order for the compression to be performed directly in the so-called raw-Bayer domain instead of in the RGB domain. As such, compression is performed before demosaicing and other processing operations that typically occur in the RGB domain — such as gamma transformation, white balance correction, colorspace transformation, etc. The coding tools were selected and the second edition of Part-1 is currently in the DIS (Draft International Standard) stage. In a nutshell, the new edition of Part-1 will normalize handling of a Bayer pattern image as a 4-component image (BGGR, RGBG, GRGB, or RGGB). It will also specify a 4-component decorrelation transform, specifically optimized for these patterns. And finally, a non-linear forward and inverse transform will be made available normatively so as to enable compression in a gamma-transformed space. Experiments have indeed shown that such a non-linear transform was beneficial, both objectively and subjectively, given the subsequent gamma transform occurring in the RGB domain later in the workflow.

Additional profiles including these new features are also developed. Final publication of these new tools is expected early 2021.

X. CONCLUSION AND OUTLOOK

JPEG XS is a new International Standard for visually lossless low-latency lightweight image coding, designed to compensate for continuously increasing bandwidth requirements in video transport links. It is a candidate technology wherever uncompressed video is used today. A simple yet efficient coding scheme allows latency and complexity to be kept very low and at the same time achieve visually lossless quality at compression ratios of up to 10:1, or even higher depending on the nature of the image or the requirements of the targeted application. Quality assessments show very good performance compared with other existing codecs, especially for multi-generation applications. In particular, in the targeted range of compression ratios (between 2 to 8

[8]IS stands for International Standard.

bpp), JPEG XS and JPEG 2000 reach similar quality levels (with JPEG 2000 slightly outperforming JPEG XS) for a given latency range, although JPEG 2000 is far more complex ($\sim$9 times more logic resources in hardware implementations), and power-consuming than JPEG XS.

Beyond the JPEG XS Core Coding System, multiple profiles and formats have been defined allowing use of this new codec within many applications. New coding tools specifically designed for efficient compression of CFA data (raw-Bayer patterns) are currently being standardized.

REFERENCES

[1] ISO/IEC JTC 1/SC 29/WG1, "Use cases and requirements for ISO/IEC 21122-1 (JPEG XS Part-1, Core Coding System) v2.1," Oct. 2020. [Online]. Available: http://ds.jpeg.org/documents/jpegxs/wg1n89030-REQ-JPEG_XS_Use_Cases_and_Requirements_v2_1.pdf

[2] M. Visca and H. Hoffmann, "HDTV production codec tests," *EBU Technical review*, p. 15, 2008. [Online]. Available: https://tech.ebu.ch/docs/techreview/trev_2008-Q3_HD-Prod-Codecs.pdf

[3] M. Cronk and C. Meyer, "Managing Delay in Live IP Production Systems," 2016 VSF October Meeting Series, Broomfield, Colorado, USA, 2016.

[4] M. Kanabus, E. Szelag, E. Rojek, and E. Poppel, "Temporal order judgement for auditory and visual stimuli," *Acta neurobiologiae experimentalis*, vol. 62, no. 4, pp. 263–270, 2002.

[5] M. C. Potter, B. Wyble, C. E. Hagmann, and E. S. McCourt, "Detecting meaning in rsvp at 13 ms per picture," *Attention, Perception, & Psychophysics*, vol. 76, no. 2, pp. 270–279, 2014.

[6] A. Willème, A. Descampe, G. Rouvroy, P. Pellegrin, and B. Macq, "Jpeg xs-based frame buffer compression inside hevc for power-aware video compression," in *Applications of Digital Image Processing XL*, vol. 10396. International Society for Optics and Photonics, 2017, p. 103960C.

[7] ISO/IEC 29170-2:2015, "AIC: Advanced image coding and evaluation methodologies — Part 2: Evaluation Procedure for nearly lossless coding," 2015.

[8] T. Ebrahimi, D. McNally, T. Bruylants, A. Willème, P. Schelkens, and B. M. Macq, "JPEG XS call for proposals subjective evaluations," in *Applications of Digital Image Processing XL*, A. G. Tescher, Ed. San Diego, United States: SPIE, Sep. 2017, p. 24. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10396/2275137/JPEG-XS-call-for-proposals-subjective-evaluations/10.1117/12.2275137.full

[9] ISO/IEC 14495-1:1998 | ITU-T Rec. T.87, "JPEG LS: Lossless and near-lossless compression of continuous-tone still images – Baseline," 1998.

[10] ISO/IEC 10918-1:1992 | ITU-T Rec. T.81, "JPEG: Digital compression and coding of continuous-tone still images — Requirements and guidelines," 1992.

[11] ISO/IEC 18477-1:2015, "JPEG XT: Scalable compression and coding of continuous-tone still images — Part 1: Scalable compression and coding of continuous-tone still images," 2015.

[12] ISO/IEC 15444-1:2019 | ITU-T Rec. T.800, "JPEG 2000 image coding system: Core coding system," 2019.

[13] Video Services Forum, "Transport of JPEG 2000 Broadcast Profile video in MPEG-2 TS over IP," https://www.videoservicesforum.org/download/technical_recommendations/VSF_TR-01_2018-06-05.pdf, 2018, accessed: 2020-11-02.

[14] ISO/IEC 15444-15:2019 | ITU-T Rec. T.814, "JPEG 2000 image coding system - part 15: High-throughput jpeg 2000," 2019.

[15] T. Edwards and M. Smith, "High Throughput JPEG 2000 for Broadcast and IP-based Applications," in *SMPTE 2020 Annual Technical Conference and Exhibition*, 2020.

[16] ISO/IEC 23008-2:2013 | ITU-T Rec. H.265, "High efficiency video coding (hevc)," 2013.

[17] "VC-2 Video Compression SMPTE ST 2042-1."

[18] SMPTE Engineering Project (ANSI), "Apple ProRes Bitstream Syntax and Decoding Process. SMPTE registered disclosure document RDD 36," 2015.

[19] Video Electronics Standards Association, "VESA Display Stream Compression (DSC)," 2014.

[20] Video Electronics Standards Association and MIPI Alliance, "VESA Display Compression VDC-M," 2018.

[21] ISO/IEC 21122-1:2019, "JPEG XS low-latency lightweight image coding system — Part 1: Core coding system," 2019.

[22] T. Richter, J. Keinert, A. Descampe, and G. Rouvroy, "Entropy Coding and Entropy Coding Improvements of JPEG XS," in *2018 Data Compression Conference*, Mar. 2018, pp. 87–96, iSSN: 2375-0359.

[23] D. Taubman and M. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards and Practice*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 2002. [Online]. Available: https://www.springer.com/gp/book/9780792375197

[24] M. Weinberger, G. Seroussi, and G. Sapiro, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Transactions on Image Processing*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[25] ISO/IEC 21122-2:2019, "JPEG XS low-latency lightweight image coding system — Part 2: Profiles and buffer models," 2019.

[26] T. Richter, J. Keinert, and S. Fößel, "Parallelization and multi-threaded latency constrained parallel coding of JPEG XS," in *Applications of Digital Image Processing XLII*, A. G. Tescher and T. Ebrahimi, Eds., vol. 11137, International Society for Optics and Photonics. SPIE, 2019, pp. 103 – 111. [Online]. Available: https://doi.org/10.1117/12.2526917

[27] IETF Payload Working Group, "RTP Payload Format for ISO/IEC 21122 (JPEG XS)," 2020.

[28] ISO/IEC 21122-3:2019, "JPEG XS low-latency lightweight image coding system — Part 3: Transport and container formats," 2019.

[29] ISO/IEC 15444-2:2004 | ITU-T Rec. T.801, "JPEG 2000 image coding system: Core coding system," 2004.

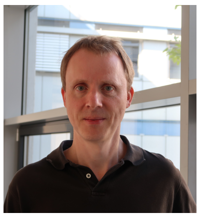
[30] Intel, "Cyclone V Product Table.pdf." [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/pt/cyclone-v-product-table.pdf

[31] intoPIX, "Datasheet of JPEG2000 Encoder - JPEG2000 Decoder IP Cores INTEL." [Online]. Available: https://www.intopix.com/Ressources/Flyers/JPEG2000%20Encoder%20-%20JPEG2000%20Decoder%20IP%20Cores%20INTEL%20intoPIX.pdf

[32] Intel, "Arria V Product Table." [Online]. Available: https://www.intel.com/content/dam/www/programmable/us/en/pdfs/literature/pt/arria-v-product-table.pdf

[33] A. Willème, S. Mahmoudpour, I. Viola, K. Fliegel, J. Pospíšil, T. Ebrahimi, P. Schelkens, A. Descampe, and B. Macq, "Overview of the JPEG XS core coding system subjective evaluations," in *Applications of Digital Image Processing XLI*, vol. 10752. International Society for Optics and Photonics, Sep. 2018, p. 107521M. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10752/107521M/Overview-of-the-JPEG-XS-core-coding-system-subjective-evaluations/10.1117/12.2323435.short

Antonin Descampe is Assistant Professor at the Media Innovation and Intelligibility Lab (MiiL) within UCLouvain, Belgium. He holds a Master degree (2003) and a Ph.D. (2008), both in Electrical Engineering, from the same university. He also holds a Master degree in Telecommunications from the Ecole Centrale Paris (ECP), France. During his PhD and post-doctoral research, he worked in the field of machine learning and image processing. In this context, he co-founded intoPIX, where he has been involved in the development and standardization of new image and video compression technologies. Active member of ISO (JPEG, MPEG), VSF and SMPTE, he has been one of the project leaders of the JPEG XS standardisation effort. His research interests include image and video processing, algorithmic accountability, and AI in the media and journalistic context.



Thomas Richter is a senior research scientist at Fraunhofer IIS Erlangen. He received his PhD in mathematical physics in the year 2000 at the TU Berlin, from where he moved to AlgoVison Technology GmbH for working on JPEG 2000. In 2002, he moved to the Multimedia Center of the TU-Berlin, and in 2007 to the Computing Center of the University of Stuttgart. Since 2017 he is with Fraunhofer IIS. Thomas Richter is a member of ISO SC29 WG01 (JPEG) since 2001 and edited multiple standards, amongst them ISO/IEC 18477 (JPEG XT) and ISO/IEC 21122-1 (JPEG XS).

Touradj Ebrahimi is professor of image processing at Ecole Polytechnique Fédérale de Lausanne (EPFL) active in teaching and research in multimedia signal processing. He is the head of Multimedia Signal Processing Group at EPFL. Since 2014, he has been the Convener of the JPEG standardization Committee which has produced a family of standards that have revolutionised the world of imaging. He represents Switzerland as the head of its delegation to JTC1 (in charge of standardization of information technology in ISO and IEC), SC29 (the body overseeing MPEG and JPEG standardization) and is a member of ITU representing EPFL. He was previously the chairman of the SC29 Advisory Group on Management until 2014. Prof. Ebrahimi is also involved in Ecma International as a member of its ExeCom. Prof. Ebrahimi serves as consultant, evaluator and expert for European Commission and other governmental funding agencies in Europe and advises a number of Venture Capital companies in Switzerland in their scientific and technical audits. He has founded several startup and spinoff companies in the past two decades, including the most recent RayShaper SA, a research startup company based in Switzerland involved in AI powered multimedia. Prof. Ebrahimi is author or co-author of over 300 scientific publications and hold a dozen invention patents. His areas of interest include image and video compression, media security, quality of experience in multimedia and AI based image and video processing and analysis. Prof. Ebrahimi is a Fellow of the IEEE and SPIE and has been recipient of several awards and distinctions, including an IEEE Star Innovator Award in Multimedia.

Siegfried Foessel received a Diploma and Ph.D. degree in Electrical Engineering from the Friedrich Alexander University in Erlangen. He is head of the department Moving Picture Technologies at Fraunhofer IIS, Germany, and professor for the technology study program at the HFF film school in Munich. In addition he is spokesperson of the Fraunhofer Business Area Digital Media. His previous research projects include: development of digital cinema cameras like ARRI D20/D21, certification test plan for DCI (Digital Cinema Initiatives), standardisation of digital cinema profiles for JPEG2000 and development of post-production tools for digital cinema (easyDCP, IMF) and immersive audio (MPEG-H). Siegfried is member and chair of various national and international standardisation bodies. He is SMPTE fellow and IEEE senior member with more than 70 publications. For FKTG (Fernseh-und Kinotechnische Gesellschaft), the German equivalent to SMPTE, he serves currently as president.

Joachim Keinert is head of group Computational Imaging and Algorithms at Fraunhofer IIS. He has diplomas from both the University of Stuttgart, Germany, and Télécom ParisTech (formerly ENST) in Paris, France. In 2009, he received a Ph.D. with honors in the domain of electronic system level design for image processing applications. His research interests focus on light-field imaging, depth acquisition and low complexity image coding. Keinert actively contributes to the standardization of the JPEG XS low complexity standard. He is experienced in development of hard- and software for image processing in both embedded and general-purpose systems.

Tim Bruylants is an experienced Software Engineer with a demonstrated history of working in both the academic world and the semiconductors industry. He received a Master in Computer Science in 2001 and became a Doctor of Engineering in 2015. With over 15 years of experience in ISO/IEC SC 29/WG1 (JPEG), he is actively working as a standardization engineer at intoPIX, and co-authored standards like JP3D, XS and XT.

Pascal Pellegrin is an expert in digital electronic design and image compression. Graduated from UCLouvain, he joined intoPIX in 2006 and is now leading the research and development team to provide innovative solutions to the audio-visual market (cinema, broadcast and ProAV). He was deeply involved in the creation of the JPEG XS compression algorithm where his experience in both electronic and imaging allowed to keep the complexity of the codec very low. He is now actively contributing to the development and the deployment of solutions based on JPEG XS with the intoPIX team.

Charles Buysschaert holds a Master's degree in Electro-mechanical engineering from the Université catholique de Louvain (UCLouvain), Belgium. He started his career as embedded software developer in the automotive and then joined the broadcast industry, working at Grass Valley. In 2009, he joined intoPIX, bringing a complementary software expertise to the development team. He has since created and grown this software team to deliver 100% software-driven products like TICO-XS coding libraries to CPU and GPU. Charles also contributed to the development of TICO algorithm and JPEG XS mainly in the software modeling aspect.

Gaël Rouvroy is a leading expert in electronic design and image algorithms. Gael's specializations are secure and reconfigurable hardware compression, cryptographic hardware, block ciphers, DRM, watermarking, image and video compression. Co-founder of intoPIX, he is today Chief Executive Officer and director of technology. He holds an electronics and mechanics engineering Master Degree and PhD from the Université catholique de Louvain (Belgium). Gael is active in several standardisation committees, has presented over 30 papers on security, watermarking, image compression and design strategies for cost effective and low power implementations and is the inventor of several patents.