electronics

MDPI

Article

Layer Selection in Progressive Transmission of Motion-Compensated JPEG2000 Video

José Carmelo Maturana-Espinosa [1], Juan Pablo García-Ortiz [1], Daniel Müller [2] and Vicente González-Ruiz [1,*]

[1] University of Almería, Ctra. Sacramento, s/n, 04120 Almería, Spain; carmelo.maturana@gmail.com (J.C.M.-E.); jp.garcia.ortiz@gmail.com (J.P.G.-O.)
[2] European Space Agency, ESTEC, P.O. Box 299, 2200 AG Noordwijk, The Netherlands; Daniel.Mueller@esa.int
* Correspondence: vruiz@ual.es

Received: 26 August 2019; Accepted: 11 September 2019; Published: 13 September 2019

Abstract: MCJ2K (Motion-Compensated JPEG2000) is a video codec based on MCTF (Motion-Compensated Temporal Filtering) and J2K (JPEG2000). MCTF analyzes a sequence of images, generating a collection of temporal sub-bands, which are compressed with J2K. The R/D (Rate-Distortion) performance in MCJ2K is better than the MJ2K (Motion JPEG2000) extension, especially if there is a high level of temporal redundancy. MCJ2K codestreams can be served by standard JPIP (J2K Interactive Protocol) servers, thanks to the use of only J2K standard file formats. In bandwidth-constrained scenarios, an important issue in MCJ2K is determining the amount of data of each temporal sub-band that must be transmitted to maximize the quality of the reconstructions at the client side. To solve this problem, we have proposed two rate-allocation algorithms which provide reconstructions that are progressive in quality. The first, OSLA (Optimized Sub-band Layers Allocation), determines the best progression of quality layers, but is computationally expensive. The second, ESLA (Estimated-Slope sub-band Layers Allocation), is sub-optimal in most cases, but much faster and more convenient for real-time streaming scenarios. An experimental comparison shows that even when a straightforward motion compensation scheme is used, the R/D performance of MCJ2K competitive is compared not only to MJ2K, but also with respect to other standard scalable video codecs.

Keywords: quantization (signal); video coding; channel allocation; scalable video coding

1. Introduction

The JPEG2000 (J2K) standard [1] is a still-image codec which also encompasses the compression of sequences of images that goes by the name Motion J2K (MJ2K). The standard relies on the J2K Interactive Protocol (JPIP) [2] to transmit J2K codestreams between client/server systems, offering a high degree of scalability (spatial, temporal and quality). These features make J2K (and its extension MJ2K) especially suitable for the management of video repositories, and for the implementation of interactive image/video streaming services [3]. In particular, JPIP has proven very effective for visualization of petabyte-scale image data of the Sun (Helioviewer Project [4,5]), allowing researchers and the general public alike to explore time-dependent image data from different space-borne observatories, interactively zoom into areas of interest and play sequences of high-resolution images at various cadences. Figure 1 shows an example of an interaction with the JHelioviewer service.

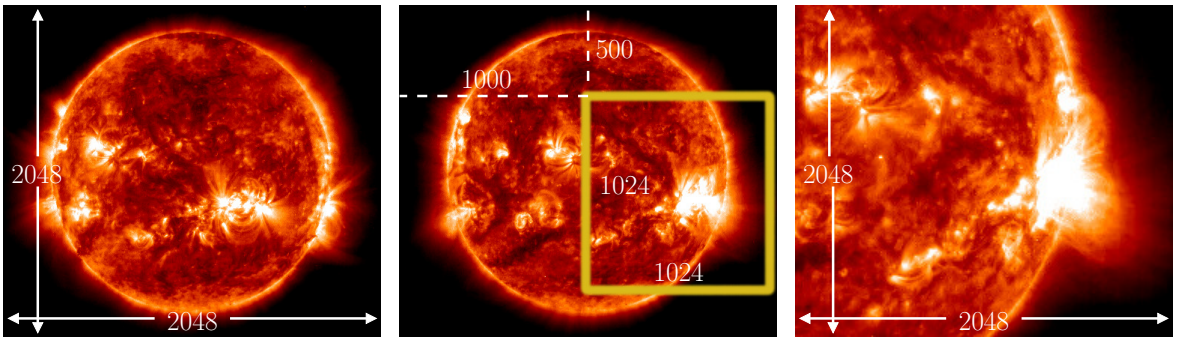

Figure 1. Different instants of a remote browsing (on a 2048 × 2048 pixels display) of a 4096 × 4096 pixels image sequence of extreme ultraviolet images of the Sun, taken by NASA's Solar Dynamics Observatory (SDO), using the JHelioviewer application. Initially, users retrieve the sequence of images with a resolution that fits in their display (left subfigure). Notice that the information of the highest spatial resolution level (4096 × 4096) is not transmitted because can not be displayed. In any moment of the transmission users can select a Window Of Interest (WOI), which in this example starts at pixel 1000 × 500 and have a (image) resolution of 1024 × 1024 (center subfigure), retrieving this WOI at the highest resolution. In the rest of the transmission (right subfigure), only the code-stream related to that WOI will be transmitted. Image credit: NASA/SDO/AIA.

MCJ2K (Motion-Compensated JPEG2000) is a combination of two fundamental stages: (1) MCTF (Motion-Compensated Temporal Filtering) and (2) J2K. Basically, MCTF is a transform that inputs a sequence of images and outputs a sequence of *MCTF-coefficients* (which will simply be called *coefs*), grouped in a collection of temporal sub-bands. Then, these coefs are compressed with J2K, resulting in a collection of J2K codestreams that can be transmitted using JPIP. The R/D performance of MCJ2K can clearly be better than that of J2K, depending on the temporal correlation among the input images. As an example, Figure 2 shows a Sun image (of a sequence) decompressed with MJ2K and MCJ2K, at similar bitrates.

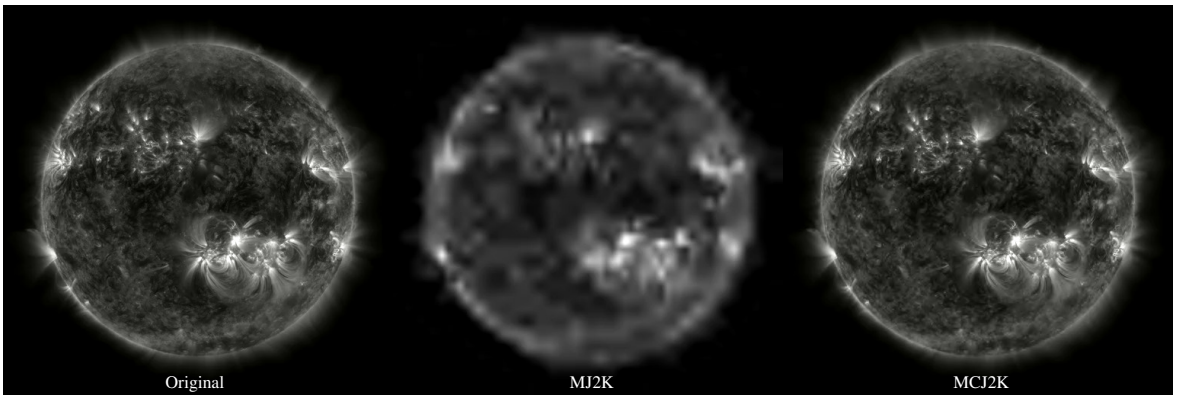

Figure 2. Reconstruction of one image of a sequence of Sun images using MJ2K and MCJ2K. Left: original Sun image with 512 × 512 pixels and a cadence of $\frac{1}{12}$ images/second. Center: same image (progressively) decompressed with MJ2K at 0.08 kbps. Right: same image (progressively) decompressed with MCJ2K at 0.04 kbps. Image credit: NASA/SDO/AIA.

MCJ2K is a straightforward extension of MJ2K, and it has been proposed previously [6]. However, the adaptation of MCJ2K to standard JPIP services, such a Helioviewer, is a novel contribution. Furthermore, two novel RA (Rate-Allocation; in this document this term refers to the action of sorting the code-stream to provide some kind of scalability, and the term "rate control" is used to decide which information is represented by the code-stream in rate-constrained scenarios)

algorithms: OSLA (Optimized Sub-band Layers Allocation) and ESLA (Estimated-slope Sub-band Layers Allocation) are herein proposed and experimentally evaluated. Both algorithms are run at post-compression time to determine an efficient progression of quality layers.

The rest of this document is structured as follows. Section 2 describes the most relevant works related to MCJ2K and RA in wavelet-based video coding. MCJ2K, OSLA and ESLA are detailed in Section 3. Section 4 presents the results of an empirical performance evaluation, and Section 5 summarizes our findings and outlines future research in Section 6.

2. Background and Related Work

The combination of MCTF and J2K has been proposed in previous works. Secker et al. use these techniques to create LIMAT [7], but no RC (Rate Control) or RA algorithms are proposed. The motion information is simply placed first, followed by the texture data.

In [6], Cohen et al. propose two ME (motion estimation)-based J2K codecs. The first is a 2D-pyramid codec with an MCTF step on each spatial level, and a closed-loop coding structure, similar to H.264/SVC [8] and HEVC [9]. The second codec is open-loop, similar to MCJ2K, but the authors do not address the problem of RA among temporal texture sub-bands and motion information.

A similar approach to [7] was designed in [10] and extended in [11] by André et al. Also Ferroukhi et al. [12] have recently proposed a similar codec based on second generation J2K. In these works, using RDO (Rate-Distortion Optimization) [13], an RC algorithm is proposed to determine the contribution of each temporal sub-band. None of these works provide an RA algorithm.

In [14] Barbarien et al. provide some interesting ideas to perform optimal RC at compression time. Before using a 2D-DWT (Discrete Wavelet Transform) [15], all the residue coefficients resulting from the MCTF stage are multiplied by a scaling factor to approximate MCTF to a unitary (energy preserving) transformation. As in [11], an optimal RC among motion and texture data is proposed using RDO. An interesting alternative was proposed in [3], where, similar to the use of quantization in hybrid video coding, a set of R/D slopes can be specified to control the composition of each quality layer (by including those layers whose R/D slopes are higher than or equal to the slope of the corresponding layer). These approaches are optimal only in linear transformation scenarios, a condition which is difficult to satisfy (as will be shown in the experimental results) when ME/MC techniques are used. The compatibility with JPIP is not studied in these works.

As previously mentioned, RA can be performed at decompression time. However, in this case, it can be implemented by the sender (server), receivers (clients) or both. FAST, proposed by Aulí-Llinàs et al. in [16] and improved by Jimenez-Rodriguez et al. in [17], is a sender-driven RA algorithm for MJ2K sequences. Another interesting MJ2K/MCJ2K sender-driven RA proposal was introduced by Naman et al., which uses Conditional Replenishment (CR) [18] and Motion Compensated (MC) [19]. In Naman's proposals, a server sends those J2K packets related to the regions that the clients should refresh to optimize the quality of the video after considering bandwidth constraints. These proposals are not fully J2K compliant at the server side (a requirement in standard JPIP services) because some kind of non-J2K-standard logic must be used.

Receiver-driven RA solutions have also been proposed in previous studies. For example, in DASH [20], clients retrieve video streams, requesting (GOP by GOP) those code-stream segments that maximize the user's QoE (Quality of Experience), and the buffer fullness. In [21], Mehrotra et al. propose an improvement of the previous approach in which clients use the R/D information of the video to select (taking into account the desired startup latency, the buffer size, and the estimated network capacity) the optimal number of quality layers (in the case of H.264/SVC), or which quality-version of each GOP (in the case of H.264/simulcasting) will be transmitted.

As in [11], in [14] the authors also propose an optimal RA among motion and texture data based on Lagrangian RDO, considering that the distortions are additive (something that can be sub-optimal in those cases where the MCTF is not linear). Such optimization minimizes the distortion for a known

bitrate, but not for any possible bitrate (note that when transmitting an image or a sequence of images, such bitrates established at compression time might not be met at decompression time).

3. MCJ2K

3.1. Codec Overview

MCJ2K is a two stages codec (see Figure 3): MCTF performs temporal filtering and MCJ2K compress the sequence of sub-bands. The resulting code-stream (see Figure 4) is a collection of compressed texture (each one composed by coefs) and motion sub-bands. MCJ2K is an open-loop "t+2D" structure. The "t" corresponds to a T-levels MCTF (a T-levels 1/3 linear 1D-DWT, denoted by MCTF^T) and the "2D" to a 2D-DWT, provided by the standard J2K codec. MCTF^T exploits the temporal redundancy and 2D-DWT, included as a part of the MJ2K compressors, the spatial redundancy. The set of MJ2K compressors inputs the coefs of each temporal sub-band generated by MCTF^T and perform entropy layered coding.

In Figure 3, s represents the original sequence and $[s]^Q$ a progressive approximation of s, reconstructed with MCJ2K using Q quality layers. MCTF^T transforms s into a collection of $T+1$ temporal texture sub-bands $\{L^T, \{H^t; 1 \leq t \leq T\}\}$, and T motion-"sub-bands" $\{M^t; 1 \leq t \leq T\}$. In Figure 3, the number of levels of MCTF is $T = 2$.

Compared to the MPEG/ITU standards, all the coefs (in our case, the images of index $i \times 2^T; i = 0, 1, \cdots$ of s) of L^T are I-type, and all the coefs of $\{H^t; 1 \leq t \leq T\}$) are B-type. More details about how MCTF has been implemented can be found in [22], and in our implementation published on GitHub (https://github.com/vicente-gonzalez-ruiz/MCTF-video-coding).

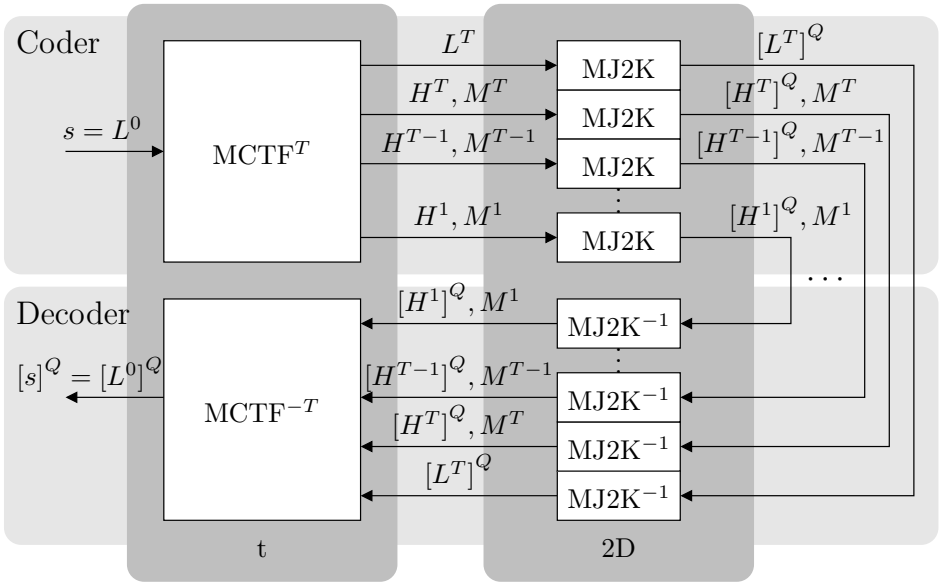

Figure 3. Codec architecture.

Figure 4 shows an example of the organization of an MCJ2K code-stream. Nine images have been compressed (although only the first six $s_0, \cdots s_5$ have been shown) using a GOP size $G = 4$ (i.e., $T = 2$ ($G = 2^T$)), except for the first GOP, which always has only one image. MCTF^2 transforms (see Figure 3) the input sequence s into 3 texture sub-bands $\{L^2, H^2, H^1\}$ and 2 motion sub-bands $\{M^2, M^1\}$. L^2 is the low-frequency texture sub-band, and represents the low-frequency temporal components of s. $\{H^2, H^1\}$ contains the high-frequency temporal components of s. $\{M^2, M^1\}$ stores a description of the motion detected in s. In Figure 4, arrows over the motion fields indicate the decoding dependencies between the coefs. When the inverse transform is applied, a succession of increasing temporal resolution levels $\{L^2, L^1, L^0\}$ are generated. By definition, $L^0 = s$.

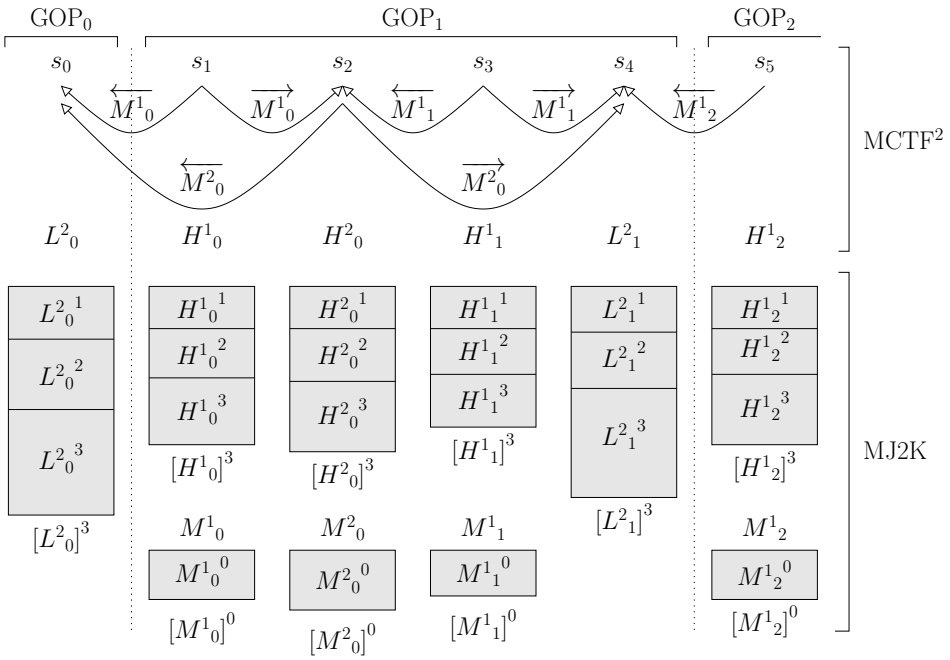

Figure 4. Example of a code-stream for MCTF^2.

MCJ2K implements a full/sub-pixel telescopic-search [23] bidirectional block-matching ME algorithm [24]. The block size B is constant (inside a coef), and a search area of A pixels is configurable. Exhaustive and logarithmic searches [25] are available. ME/MC operations are performed at the maximum spatial resolution of the sequence. This design decision, which is convenient for a progressive-quality visualization of the full-resolution video, implies that the inverse motion compensation process must be performed at the maximum resolution to avoid a drift error [26], when a reduced resolution of the images is decoded. Obviously, in this case, the decoder increases the computing requirements, but this does not significantly increase the memory usage if all the blocks are not processed in parallel. As an advantage, the quality of the reconstructions is higher than for the case in which the ME/MC state is performed at a lower resolution because the motion information is always used with the accuracy used at the compression, which can be sub-pixel.

Motion data is temporally and spatially decorrelated, and lossless MJ2K-compressed as a sequence of 4-component (2 vectors per macro-block) single-layer ($Q = 1$) images. (Usually, the use of approximate motion information generates severe artifacts in the reconstructed images and increases the non-linearity of the codec. Therefore, only one quality layer and lossless coding was used for the motion sub-bands.) The decorrelation process uses an algorithm in which, when no motion data is received, the inverse MCTF process supposes that all the motion vectors are zero. Thus (in the transmission process), when the decoder knows M^T, then it is supposed that the motion vectors of M^{T-1} are half of the value of M^T, and this linear prediction is used for the remaining temporal resolution levels [27].

3.2. Bitrate Control

RC is performed at compression time. In the MJ2K stage, each coef of each texture sub-band is J2K-compressed, producing a layered variable-length code-stream (in the Figure 4, $Q = 3$ quality layers). Let ${S_i}^q$ be the q-th quality layer of the compressed representation of the coef S_i of sub-band S, and $(S_i)^q$ the quality (i.e., the decrease in distortion) provided by ${S_i}^q$ in the progressive reconstruction of S_i. Assuming that the distortion metric is additive, we define

$$[S_i]^q = \sum_{j=1}^{q} (S_i)^j, \tag{1}$$

which is the quality of the reconstruction of the coef S_i using q layers. (In this notation, the first quality layer, in the layers decoding order, has the index 1.) We define the q-th R/D slope of coef S_i as

$$\lambda_{S_i{}^q} = \frac{[S_i]^q - [S_i]^{q-1}}{l(S_i{}^q)} = \frac{(S_i)^q}{l(S_i^q)}, \tag{2}$$

where $l(S_i{}^q)$ represents the length of $S_i{}^q$.

Owing to how the R/D slopes are chosen in the MJ2K stage, it holds that for any two different coefs i and j of sub-band S

$$\lambda_{S_i{}^q} = \lambda_{S_j{}^q} \forall q \in \{1, \cdots, Q\}. \tag{3}$$

We define a *sub-band layer* (SL) S^q (of motion (In the case of the motion, the definition is identical, but there is only one quality layer.) or texture) as the collection of quality layers

$$S^q = \{S_i{}^q, i = 0, \cdots, 2^T - 1\}. \tag{4}$$

For example, in Figure 4, SL $L^{2^1} = \{{L^2}_0{}^1, {L^2}_1{}^1\}$ and SL $M^{1^0} = \{{M^1}_0{}^0, {M^1}_1{}^0\}$.

Equation (3) has two implications: (1), in general, the total length of the code-stream of each coef will be different (depending on its content), and (2) the bitrate allocation is optimal for each sub-band layer [28].

The q-th R/D slope of SL S^q is defined as

$$\lambda_{S^q} = \frac{[S]^q - [S]^{q-1}}{l(S^q)} = \frac{(S)^q}{l(S^q)}, \tag{5}$$

where $l(S^q)$ represents the length of SL S^q, $[S]^q$ the quality of the GOP obtained after decompressing q layers, and $(S)^q$ the quality provided by the SL S^q.

3.3. Post-Compression R/D Allocation

RA is typically performed at decompression time. In accordance with Part 9, Section C.4.10 of the J2K standard [2], JPIP clients can request J2K images by quality layers. Moreover, as previously shown in [29], it is also possible to perform JPIP request for a range of images. Therefore, by extension, the JPIP standard can also be used for retrieving complete sub-band layers using a single JPIP request. For example (see Figure 4), if $T = 2$, we decompose a sequence in 3 temporal sub-bands, and the sub-band layer H^{2^1} has, for each GOP, only one coef ${H^2}_0$ with two quality layers $\{{H^2}_0{}^1, {H^2}_1{}^1\}$, which would be that which is requested by a client to retrieve the sub-band layer H^{2^1}.

It is easy to see that the SLs in a MCJ2K code-stream are

$$\begin{array}{lllll} L^{T^1}, & H^{T^1}, & H^{T-1^1}, & \cdots, & H^{1^1}, \\ L^{T^2}, & H^{T^2}, & H^{T-2^2}, & \cdots, & H^{1^2}, \\ \vdots & \vdots & \vdots & & \vdots \\ L^{T^Q}, & H^{T^Q}, & H^{T-1^Q}, & \cdots, & H^{1^Q}, \\ & M^T, & M^{T-1}, & \cdots, & M^1, \end{array} \tag{6}$$

and that there are $Q(T+1)$ SLs in this set, which is also the number of optimal truncation points of a MCJ2K code-stream.

At decompression time, the order in which the SLs are retrieved from the JPIP server should minimize the R/D curve, for any bitrate. For this task, we propose the following two approaches.

3.3.1. Optimized SL Allocation (OSLA)

Starting at L^{T^1}, the optimal order of the remaining SLs of a GOP can be determined by applying Equation (5) to each feasible SL, and sorting them by slope. Thus, after retrieving L^{T^1} (which always contributes to the quality of the reconstruction more than any other SL), several alternatives $\{M^T, M^{T-1}, \cdots, M^1, H^{T^1}, H^{T-1^1}, \cdots, H^{1^1}\}$ should be checked to determine the next SL with the highest possible contribution. Considering that $\lambda_{M^T} > \lambda_{M^t}, \forall t \in \{T-1, \cdots, 1\}$, for example, if $\lambda_{M^T} > \lambda_{H^{t^1}}, \forall t \in \{T, \cdots, 1\}$, the following SL to decode should be M^T and the next set of alternatives would be $\{M^{T-1}, H^{T^1}, H^{T-1^1}, \cdots, H^{1^1}\}$. Otherwise, if for example, $\lambda_{H^{T^1}} > \lambda_{M^t}$ and $\lambda_{H^T} > \lambda_{H^t}, \forall t = T-1, \cdots, 1$, after L^{T^1} the next SL to decode should be H^{T^1}, and the current set of feasible SLs of would be $\{M^T, H^{T^2}, H^{T-1^1}, \cdots, H^{1^1}\}$. Notice also that other SLs could follow L^{T^1}, such as L^{T^2}.

This idea was implemented in the OSLA algorithm (see Algorithm 1). For each GOP, the input sequence of $Q(T+1)$ SLs is S sorted in descending order by their R/D slopes to reconstruct the GOP (see Equation (5)). The output list Λ of sorted-by-slope SLs can be stored in a COM segment of the header of the coef of L^T (the SL L^{T^1} is always the first in Λ). Next, JPIP clients retrieve the quality layers of each coef of the GOP in the order specified in Λ.

Algorithm 1: OSLA algorithm.

1. `for each` GOP:
2. $\Lambda = []; i = 0$
3. $\Lambda[i++] =$ `input` L^{T^1}
4. $S =$ `input` $\{L^{T^2}, M^T, H^{T^1}, \cdots, H^{1^1}\}$
5. `while` $S \neq \emptyset$:
6. $s = \arg\max_{s \in S} (\lambda_s \geq \lambda_{s'}, \forall s' \in S)$
7. $\Lambda[i++] = s$
8. $S = S \setminus \{s\}$
9. `if` $s = M_i$:
10. $S = S \cup \{M^{i-1}\}$
11. `else if` $s = L^{T^i}$:
12. $S = S \cup \{L^{T^{i-1}}\}$
13. `else /*` $s = H^{t^i}$ `*/`:
14. $S = S \cup \{H^{t^{i-1}}\}$
15. `output` Λ

3.3.2. Estimated-slope SLs Allocation (ESLA)

Ignoring any possible effect of the non-linear behavior of the ME/MC stage, our implementation of MCTF approximates to a biorthogonal transform and, therefore, each sub-band $\{L^T, H^T, \cdots, H^1\}$ contributes with a different amount of energy to the reconstruction of the sequence. This can easily be verified by comparing the energy that the different coefs of each temporal sub-band contribute to reconstruction of the sequence [30]. How much energy a coef must contribute to the code-stream to approximate MCTF to an orthonormal (energy preserving) transform is represented by attenuation values (see Table 1)

$$\alpha_{H^t} = \frac{E(L^T)}{E(H^t)}, \tag{7}$$

where $E(\cdot)$ represents the signal energy. These attenuations are empirical, specifically determined for the 1/3 ME-driven DWT implemented in our codec (for a different transform, other values would be obtained).

Table 1. L_2-norm (energy) of the MCTF basis functions for the temporal sub-bands, expressed as attenuation values.

MCTF^1		MCTF^2		MCTF^3		MCTF^4		MCTF^5		MCTF^6		MCTF^7	
H^t	a_t	H^t	a_t	H^t	a_t	H^t	a_t	H^t	a_t	H^t	a_t	H^t	a_t
H^1	1.246	H^2	1.250	H^3	1.160	H^4	1.088	H^5	1.046	H^6	1.023	H^7	1.012
		H^1	1.865	H^2	2.122	H^3	2.130	H^4	2.079	H^5	2.043	H^6	2.023
				H^1	3.167	H^2	3.888	H^3	4.061	H^4	4.063	H^5	4.039
						H^1	5.802	H^2	7.431	H^3	7.936	H^4	8.031
								H^1	11.089	H^2	14.522	H^3	15.688
										H^1	21.669	H^2	28.707
												H^1	42.835

The ESLA algorithm incorporates these attenuations to scale the R/D slopes of each SL of each GOP, when these slopes have been determined taking into consideration only the reconstruction of the corresponding coef (not the reconstruction of the full GOP, as OSLA does (notice that for this reason, OSLA does not need to use such attenuations). Thus, for example, an R/D slope for a quality layer of a coef of the sub-band H^3 resulting from an MCTF^5 is divided by 4.061. In cases where there is more than one coef in a temporal sub-band, as in this example, the average of all the scaled slopes is used to determine the contribution of the corresponding SL.

This idea was implemented in ESLA (see Algorithm 2). As in OSLA, for each GOP, the input sequence of $Q(T+1)$ SLs S is sorted in descending order by their estimated R/D slope, but now the slopes of the SLs are computed directly as a weighted average of the R/D slopes of the quality layers of the corresponding coefs. If these slopes are predefined (the compression of the coefs uses the same slopes set of Q slopes for all the coefs), ESLA can be run at the receiver side without sending any R/D information. This means that the JPIP client can determine the order of SLs Λ for all the GOPs of the sequence after receiving only T, Q, and knowing the sub-band attenuations (Table 1), which does not depend on the sequence. For this reason, ESLA is more suitable than OSLA for real-time streaming scenarios.

Algorithm 2: ESLA algorithm.

1. `for each` GOP:
2. $\quad \Lambda = [\,]; i = 0$
3. $\quad$ `for each` $q \in \{1, \cdots, Q\}$:
4. $\quad\quad \Lambda[i++] =$ `input`$\{\lambda_{H^{Tq}}, \cdots, \lambda_{H^{1q}}\}$
5. $\quad$ `for each` $\lambda_k \in \Lambda$:
6. $\quad\quad \lambda_k = \lambda_k / \alpha_k$
7. $\quad \Lambda[i++] =$ `input`$\{\lambda_{L^{T1}}, \cdots, \lambda_{L^{1Q}}\}$
8. $\quad$ `sort_in_descending_order` Λ
9. $\quad$ `output` Λ

4. Evaluation

The performance of MCJ2K was evaluated for different working configurations and compared to previous proposals.

4.1. Materials and Methods

Several test videos were used for our evaluation:

1. `Mobile` (http://trace.eas.asu.edu/yuv/mobile/mobile_cif.7z) (YUV 4:2:0, 352 × 288 pixels, 30 Hz), a low-resolution video with complex movement.
2. `Container` (http://trace.eas.asu.edu/yuv/container/container_cif.7z) (YUV 4:2:0, 352 × 288 pixels, 30 Hz), a low-resolution video with simple movement.

3. `Crew` (ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/CREW_704x576_60_orig_01_yuv.zip) (YUV 4:2:0 704 × 576 pixels, 60 Hz), a medium-resolution video with complex movement.
4. `CrowdRun` (ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/) (YUV 4:2:0, 1920 × 1080 pixels, 50 Hz), a high-resolution video with a high degree of movement.
5. `ReadySetGo` (http://ultravideo.cs.tut.fi/video/ReadySetGo_3840x2160_120fps_420_8bit_YUV_RAW.7z) (YUV 4:2:0 3840 × 2160 pixels, 120 Hz), a high-resolution high degree of movement.
6. `Sun` (http://helioviewer.org/jp2/AIA/2015/06/01/131/) (monochromatic, due to represent only one frequency of the spectrum radiated by the Sun, 4096 × 4096 pixels, 1/30 Hz) a sequence of images of the Sun with only small-scale frame-to-frame motion (which is, however, complex to predict due to the random motions of magnetic flux concentrations in the Sun's photosphere).

In all experiments, 129 images were compressed, and the search range for ME was 4 pixels using full-pixel accuracy of ($A = 0$). The block size (B) was 32 × 32 for `Mobile`, `Container` and `Crew`, 64 × 64 for `CrowdRun` and `ReadySetGo`, and 128 × 128 for `Sun`. The parameters used for compressing the coefs and the images were 5 levels for the DWT, no precinct partition and code-blocks of 64 × 64 coefficients. The number of quality layers (Q) was 8, which provides a good tradeoff between the compression performance and the granularity for the rate-allocation. In the case of the motion data, $Q = 1$ and no DWT were used.

4.2. Impact of Motion Compensation

Figure 5 shows the performance of MCJ2K compared to MJ2K for different GOP sizes. Each video was compressed once and decompressed progressively, sorting the subband layers using OSLA. MCJ2K was in most of cases superior to MJ2K, depending on the temporal correlation found in each video. For example, MCTF is very efficient in `Container`, in which it can be seen that, for example, at 300 Kbps, MCJ2K is about 10 dB better than MJ2K. However, in the case of `ReadySetGo`, in which MCTF is not able to generate accurate predictions, the use of a GOP size larger than 4 does not increase the quality of the reconstructions. Therefore, the GOP size has a high impact on the performance of MCJ2K and is a parameter that should be optimized for every video sequence. Nevertheless, it can be expected that GOP sizes of 4 and 8 should work well for most sequences. We would like to highlight here that the MC model used in MCJ2K is very basic. More advanced predictors, such as those used in the last video coding standards cited earlier, would facilitate the use of larger GOP sizes and, therefore, higher compression ratios.

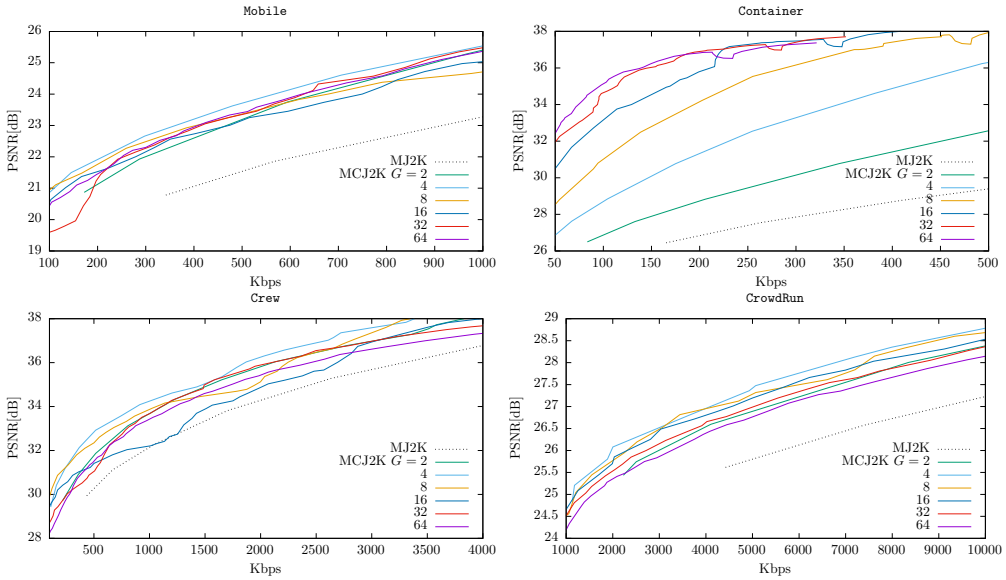

Figure 5. *Cont.*

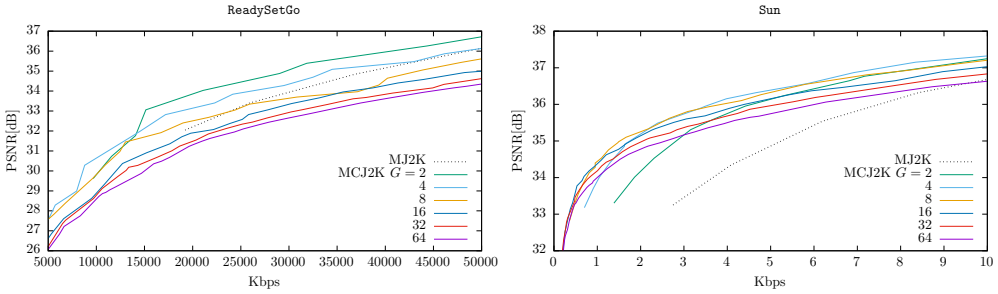

Figure 5. MCJ2K (OSLA) vs. MJ2K for different GOP sizes and sequences.

4.3. MCJ2K (Using OSLA or ESLA) vs. MJ2K

Using the information provided by the previous experiments, we selected a suitable GOP size for each sequence and compared the performance of OSLA and ESLA, respect to MJ2K. The results are shown in Figure 6. As can be seen, the performance of both RA algorithms is similar, which means that although the MCTF process used by MCJ2K is not linear, a reasonable prediction of the impact of the SLs can be made in ESLA, which runs much faster than OSLA. For this reason, in the following experiments only ESLA will be used.

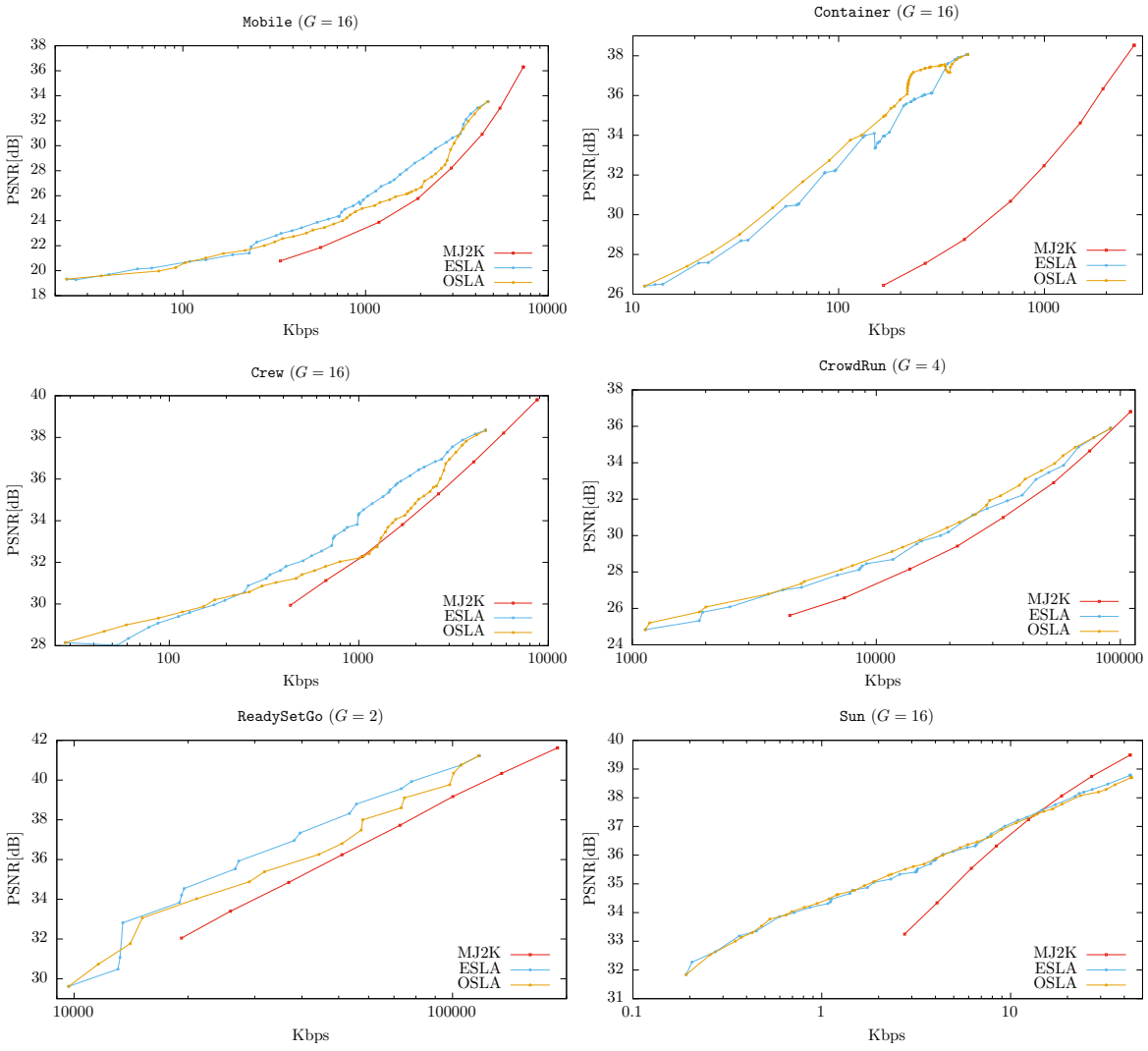

Figure 6. MCJ2K (using OSLA or ESLA) vs. MJ2K for different sequences.

4.4. MCJ2K vs. Other Video Codecs

Figure 7 shows the compression performance of MCJ2K (using ESLA and optimized compression parameters found in previous experiments) and other standard video codecs. Dashed lines represent a non-embedded decoding, while solid lines, a progressive decoding provided by scalable codecs. As can be seen, compared with non-scalable video codecs (which generally produce videos with a better R/D ratio than scalable video codecs), such as HEVC (https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware using `trunk/cfg/encode_randomaccess_main.cfg`) or AVC (http://www.videolan.org/developers/x264.html using `-profile high-preset placebo-tune psnr`), MCJ2K needs approximately 50% more data to achieve the same quality, but this difference is much smaller when it is compared with SHVC (https://hevc.hhi.fraunhofer.de/svn/svn_SHVCSoftware/ using `branches/SHM-dev/cfg/encoder_randomaccess_scalable.cfg` and `branches/SHM-dev/cfg/misc/layers8.cfg`) where MCJ2K produces better results for some of the test videos (even using a very basic MCTF scheme). In the case of MPEG-2 (http://linux.die.net/man/1/mpeg2enc, a codec that implements an MCTF scheme similar to the used in MCJ2K), MCJ2K outperforms it consistently. These results are consistent with the ME prediction model used in MCJ2K, which is not the focus of this research work.

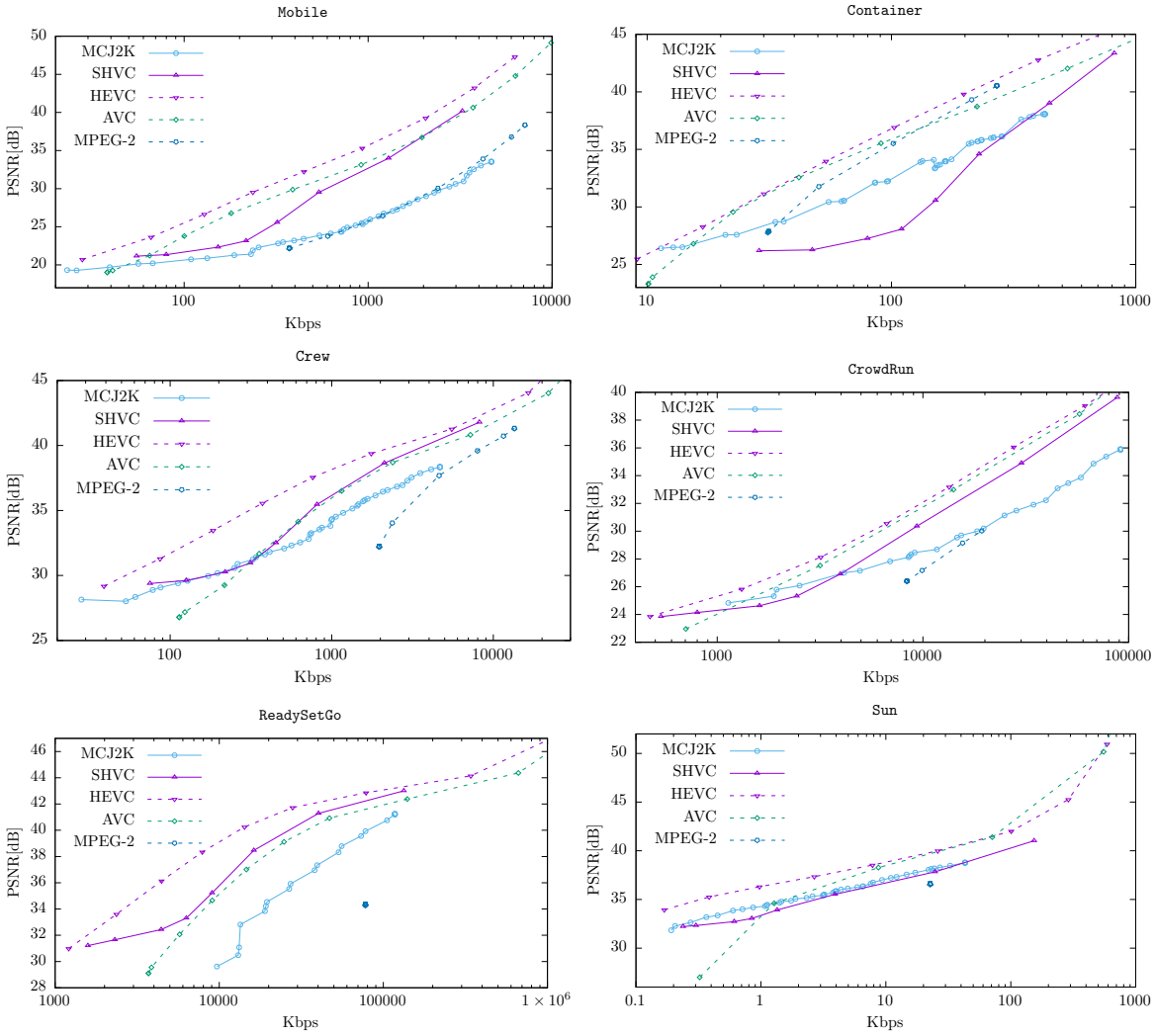

Figure 7. MCJ2K (using ESLA) vs. other codecs for different sequences.

5. Conclusions

This work presents MCJ2K, a straightforward extension (JPIP compatible) of the MJ2K standard that can be used to exploit the temporal redundancy of the sequences of images. Two different rate-allocation algorithms (OSLA and ESLA) are proposed to be used in a streaming scenario where quality scalability is used, generating a reconstruction at the receiver proportional to the amount of decoded data. After analyzing MCJ2K and the proposed rate-allocation algorithms, the following can be concluded:

1. The compression ratio obtained by MCJ2K is superior to MJ2K if enough time redundancy can be exploited in the MCTF stage. Our experiments show that the quality of the reconstructions can be up to 10 dB in terms of PSNR.
2. The increment in the compression ratio provided by OSLA compared to ESLA is small. Considering that the RD performance of OSLA is better than ESLA when the MCTF process is not linear, we conclude (1) that MCDWT is almost linear, and (2) that the position of the motion information in the progression generated by ESLA is near optimal.
3. Considering that ESLA requires less computational resources than OSLA, and that ESLA needs to be run only at the receiver, ESLA should be the rate-allocation algorithm used by default in MCJ2K.
4. MCJ2K implies recompressing the sequences of images but not modifying the JPIP servers at all. Only the JPIP clients need to implement the logic needed by MCJ2K.
5. The compression ratio obtained by MCJ2K is comparable to SHVC, when the movement of the video can be modeled using our ME proposal. However, the quality granularity and the range of decoding bitrates is higher in MCJ2K, which makes MCJ2K more suitable than SHVC for streaming scenarios.
6. In MCJ2K the GOP size G significantly affects the RD performance. G should be high if the temporal correlation of the video can be removed by the MCTF stage, and vice versa.
7. Compared to the state-of-the-art non-scalable video compressors (such as HEVC), MCJ2K require more bitrate because HEVC use more effective ME schemes than MCJ2K (an aspect out of the scope of this paper). However, at very low bitrates this gap is usually small.

6. Future Research

Future lines of work should be focused on:

1. Like the rest of video codecs based on MC, MCJ2K has a cost in terms of temporal scalability. A study on the number of bytes required for obtaining the same quality in both codecs, MJ2K and MCJ2K, when only one image of the sequence is decoded could prove worthwhile, especially in interactive browsing systems such as Helioviewer.
2. Find a quality scalable representation of the motion data. Such a contribution should reduce the minimal number of bytes required for rendering the image sequence.
3. The use of more accurate MCTF schemes should increase the compression ratios.
4. The use of encoding schemes where the motion information can be estimated at the decoder (to avoid being sent as a part of the code-stream). This can be carried out in those contexts where the large-scale motion is predictable, such as image sequences of the Sun, whose rotation rate is stable and well known.
5. How MCJ2K affects the spatial/WOI scalability provided by the J2K standard.

Author Contributions: conceptualization, J.C.M.-E., V.G.-R., J.P.G.-O. and D.M.; methodology, V.G.-R. and J.P.G.-O.; software, J.C.M.-E. and V.G.-R.; validation, J.C.M.-E., V.G.-R., J.P.G.-O. and D.M.; investigation, J.C.M.-E.; resources, D.M.; data curation, J.C.M.-E. and V.G.-R.; writing–original draft preparation, J.C.M.-E. and V.G.-R.; writing–review and editing, J.C.M.-E. and V.G.-R.; supervision, V.G.-R.; project administration, V.G.-R.; funding acquisition, V.G.-R.

Funding: Work supported by the Spanish Ministry of Economy and Competitiveness (RTI2018-095993-B-100) and Junta de Andalucía (P10-TIC-6548), in part financed by the European Regional Development Fund (ERDF) and Campus de Excelencia Internacional Agroalimentario (ceiA3).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ISO. *Information Technology-JPEG 2000 Image Coding System-Core Coding System*; ISO/IEC 15444-1:2004; ISO: Geneva, Switzerland, 2004.
2. ITU. Information Technology-JPEG 2000 Image Coding System: Interactivity Tools, APIs and Protocols. Available online: http://www.itu.int/rec/T-REC-T.808-200501-I (accessed on 26 August 2019).
3. Bilgin, A.; Marcellin, M. JPEG2000 for digital cinema. In Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS), Island of Kos, Greece, 21–24 May 2006; pp. 4–3881, doi:10.1109/ISCAS.2006.1693475. [CrossRef]
4. Müller, D.; Dimitoglou, G.; Caplins, B.; Ortiz, J.P.; Wamsler, B.; Hughitt, K.; Alexanderian, A.; Ireland, J.; Amadigwe, D.; Fleck, B. JHelioviewer: Visualizing large sets of solar images using JPEG 2000. *Comput. Sci. Eng.* **2009**, *11*, 38–47. [CrossRef]
5. Müller, D.; Nicula, B.; Felix, S.; Verstringe, F.; Bourgoignie, B.; Csillaghy, A.; Berghmans, D.; Jiggens, P.; García-Ortiz, J.; Ireland, J.; Zahniy, S.; Fleck, B. JHelioviewer-Time-dependent 3D visualisation of solar and heliospheric data. *Astron. Astrophys.* **2017**, *606*, A10. [CrossRef]
6. Cohen, R.; Woods, J. Resolution scalable motion-compensated JPEG 2000. In Proceedings of the 2007 15th International Conference on Digital Signal Processing, Cardiff, UK, 1–4 July 2007; pp. 459–462.
7. Secker, A.; Taubman, D. Lifting-based Invertible Motion Adaptive Transform (LIMAT) framework for highly scalable video compression. *IEEE Trans. Image Process.* **2003**, *12*, 1530–1542, doi:10.1109/TIP.2003.819433. [CrossRef] [PubMed]
8. Schwarz, H.; Marpe, D.; Wiegand, T. Overview of the scalable video coding extension of the H. 264/AVC standard. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 1103–1120. [CrossRef]
9. Sullivan, G.; Ohm, J.; Han, W.J.; Wiegand, T. Overview of the High Efficiency Video Coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **2012**, *22*, 1649–1668, doi:10.1109/TCSVT.2012.2221191. [CrossRef]
10. Andre, T.; Cagnazzo, M.; Antonini, M.; Barlaud, M. JPEG2000-compatible scalable scheme for wavelet-based video coding. *EURASIP J. Image Video Process.* **2007**, *2007*, 1–11, doi:10.1155/2007/30852. [CrossRef]
11. Cagnazzo, M.; Castaldo, F.; Andre, T.; Antonini, M.; Barlaud, M. Optimal motion estimation for wavelet motion compensated video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 907–911, doi:10.1109/TCSVT.2007.897110. [CrossRef]
12. Ferroukhi, M.; Ouahabi, A.; Attari, M.; Habchi, Y.; Taleb-Ahmed, A. Medical video coding based on 2nd-generation wavelets: Performance evaluation. *Electronics* **2019**, *8*, 88. [CrossRef]
13. Sullivan, G.J.; Wiegand, T. Rate-distortion optimization for video compression. *IEEE Signal Process. Mag.* **1998**, *15*, 74–90. [CrossRef]
14. Barbarien, J.; Munteanu, A.; Verdicchio, F.; Andreopoulos, Y.; Cornelis, J.; Schelkens, P. Motion and texture rate-allocation for prediction-based scalable motion-vector coding. *Signal Process. Image Commun.* **2005**, *20*, 315–342. [CrossRef]
15. Ouahabi, A. *Signal and Image Multiresolution Analysis*; Wiley Online Library: Hoboken, NJ, USA, 2012.
16. Aulí-Llinàs, F.; Bilgin, A.; Marcellin, M. FAST Rate Allocation Through Steepest Descent for JPEG2000 video transmission. *IEEE Trans. Image Process.* **2011**, *20*, 1166–1173, doi:10.1109/TIP.2010.2077304. [CrossRef] [PubMed]
17. Jiménez-Rodríguez, L.; Aulí-Llinàs, F.; Marcellin, M. FAST rate allocation for JPEG2000 video transmission over time-varying channels. *IEEE Trans. Multimed.* **2013**, *15*, 15–26, doi:10.1109/TMM.2012.2199973. [CrossRef]
18. Naman, A.; Taubman, D. JPEG2000-based Scalable Interactive Video (JSIV). *IEEE Trans. Image Process.* **2011**, *20*, 1435–1449, doi:10.1109/TIP.2010.2093905. [CrossRef] [PubMed]
19. Naman, A.; Taubman, D. JPEG2000-Based Scalable Interactive Video (JSIV) with motion compensation. *IEEE Trans. Image Process.* **2011**, *20*, 2650 –2663, doi:10.1109/TIP.2011.2126588. [CrossRef] [PubMed]

20. ISO/IEC 23009-1:2012 Information Technology—Dynamic Adaptive Streaming over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats. Available online: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=57623 (accessed on 26 August 2019).
21. Mehrotra, S.; Zhao, W. Rate-distortion optimized client side rate control for adaptive media streaming. In Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP), Rio de Janeiro, Brazil, 5–7 October 2009; pp. 1–6, doi:10.1109/MMSP.2009.5293246. [CrossRef]
22. Secker, A.; Taubman, D. Motion-compensated highly scalable video compression using an adaptive 3D wavelet transform based on lifting. In Proceedings of the IEEE International Conference on Image Processing, Thessaloniki, Greece, 7–10 October 2001; Volume 2, pp. 1029–1032.
23. Suguri, K.; Minami, T.; Matsuda, H.; Kusaba, R.; Kondo, T.; Kasai, R.; Watanabe, T.; Sato, H.; Shibata, N.; Tashiro, Y.; others. A real-time motion estimation and compensation LSI with wide search range for MPEG2 video encoding. *IEEE J. Solid-State Circuits* **1996**, *31*, 1733–1741. [CrossRef]
24. Wu, S.; Gersho, A. Joint estimation of forward/backward motion vectors for MPEG interpolative prediction. *IEEE Trans. Image Process.* **1994**, *3*, 684–687. [CrossRef] [PubMed]
25. Hsieh, C.; Liu, Y. Fast Search Algorihtms for Vector Quantization of Images Using Multiple Triangle Inequalities and Wavelet Transform. *IEEE Trans. Image Proc.* **2000**, *9*, 321–328. [CrossRef] [PubMed]
26. Mokry, R.; Anastassiou, D. Minimal error drift in frequency scalability for motion-compensated DCT coding. *IEEE Trans. Circuits Syst. Video Technol.* **1994**, *4*, 392–406. [CrossRef]
27. Andreopoulos, Y.; van der Schaar, M.; Munteanu, A.; Barbarien, J.; Schelkens, P.; Cornelis, J. Fully-scalable wavelet video coding using in-band motion compensated temporal filtering. In Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP'03), Hong Kong, China, 6–10 April 2003; Volume 3, pp. 417–420.
28. Dagher, J.; Bilgin, A.; Marcellin, M. Resource-constrained rate control for Motion JPEG2000. *IEEE Trans. Image Process.* **2003**, *12*, 1522–1529, doi:10.1109/TIP.2003.819228. [CrossRef] [PubMed]
29. Sánchez-Hernández, J.; García-Ortiz, J.; González-Ruiz, V.; Müller, D. Interactive streaming of sequences of high resolution JPEG2000 images. *IEEE Trans. Multimed.* **2015**, *17*, 1829–1838. [CrossRef]
30. Xiong, R.; Xu, J.; Wu, F.; Li, S. Adaptive MCTF based on correlation noise model for SNR scalable video coding. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME), Toronto, ON, Canada, 9–12 July 2006; pp. 1865–1868.