AV1 and VVC Video Codecs: Overview on Complexity Reduction and Hardware Design

MARCEL CORRÊA [1,2], **MÁRIO SALDANHA** [2], **ALEX BORGES** [2] **(Graduate Student Member, IEEE),**
GUILHERME CORRÊA [3] **(Senior Member, IEEE), DANIEL PALOMINO** [4] **(Member, IEEE),**
MARCELO PORTO [3] **(Senior Member, IEEE), BRUNO ZATT** [3] **(Senior Member, IEEE),**
AND LUCIANO AGOSTINI [3] **(Senior Member, IEEE)**

[1] DEPEX, Federal Institute of Education, Science and Technology Sul-rio-grandense (IFSul), Bagé 96418-400, Brazil
[2] PPGC, Federal University of Pelotas (UFPel) , Pelotas 96010-610, Brazil
[3] CDTec, Federal University of Pelotas (UFPel), Pelotas 96010-610, Brazil
[4] CEng, Federal University of Pelotas (UFPel), Pelotas 96010-450., Brazil
This article was recommended by Associate Editor D. Galayko.

CORRESPONDING AUTHOR: M. CORRÊA (e-mail: mmcorrea@inf.ufpel.edu.br)

ABSTRACT This article presents an extensive review of the state-of-the-art system-level solutions featuring complexity reduction and/or dedicated hardware designs for the AV1 and VVC video coding formats. These formats introduced several novel coding techniques compared to their predecessors to improve the coding efficiency at the cost of a significant computational cost. In this article, we discuss the main novelties of AV1 and VVC in each coding module, including block partitioning, intra and inter prediction, transform, entropy coding, and in-loop filters. Then, we present the main published works focusing on complexity reduction and hardware designs for AV1 and VVC. Most of the complexity reduction solutions target the complex and flexible block partitioning structures of these encoders to provide a better tradeoff between coding efficiency and complexity reduction whereas the hardware designs focus on the challenge of implementing the new coding tools to attend real-time processing of high-definition videos. Even with the presented works reaching impressive results, these research fields remain opened for innovative contributions, as discussed in this article.

INDEX TERMS AV1, VVC, complexity reduction, hardware design, system-level solution, video codec.

I. INTRODUCTION

THE LIMITS of the telecommunication infrastructures available bandwidth are being pushed each day by the digital video traffic over the Internet due to the continuous growth in consumption of products that rely heavily on videos, such as social media, streaming services and video conferencing platforms.

According to Cisco, the digital video traffic grew 29% annually over the past four years, and this type of traffic is expected to reach 325 Exabytes monthly in 2022, representing 82% of the global Internet traffic [1]. It is also expected that in 2022, of all video traffic, 22.3% will be in UHD (Ultra High Definition) resolutions and 56.8% in

HD (High Definition) [1]. This will lead to an even bigger impact on the near future since video definition increase leads to a multiplicative effect on the data traffic.

The AOMedia Video 1 (AV1) [2], [3] and the Versatile Video Coding (VVC) [4], [5] are the next-generation video coding formats developed to tackle these problems. AV1 was released in June 2018 by the Alliance for Open Media (AOMedia) industry consortium, as the successor of VP9 [6]. It was developed with the main goal of being the state-of-the-art royalty-free video format, motivated by the fact that the major success of the Internet is that its core technologies are open and freely implementable, and digital videos are a central part of the Internet experience nowadays. The VVC standard
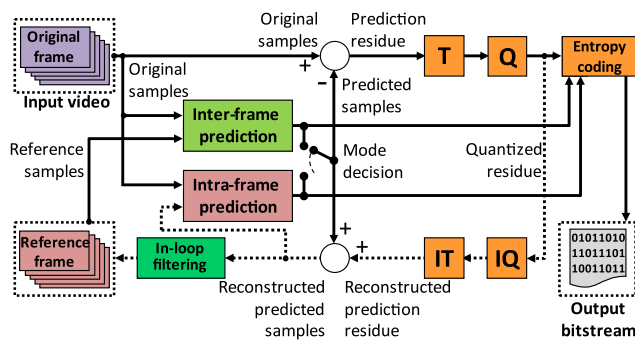
FIGURE 1. Block diagram of a typical hybrid video encoder.

FIGURE 2. AV1 10-way block partition tree structure.

FIGURE 3. The six partitions allowed in the VVC QTMT.

was released in July 2020. It comes from a long line of successful video coding standards defined by a joint effort of the Moving Picture Experts Group (MPEG) and the Video Coding Experts Group (VCEG) that includes High Efficiency Video Coding (HEVC) [7], H.264 Advanced Video Coding (AVC) [8], and MPEG-2 [9].

Several new coding tools were developed and enhanced for AV1 and VVC to deal with the new requirements of video applications and provide high coding efficiency. These improvements include larger block sizes, flexible block partitioning structures, a higher number of intra prediction modes, the support of affine modes for inter prediction, more transform sizes and types, improved implementations of quantization and entropy coding, more in-loop filters, and others.

Even though next generations of video codecs (encoder/decoder) such as AV1 and VVC can achieve a satisfactory rate-distortion performance for current video content, this efficiency comes at the cost of a high computational effort. This makes video coding a very difficult task for software solutions when real-time processing and high resolutions are desired. Besides, in a world dominated by battery-powered video-enabled embedded devices, efficient complexity reduction solutions and dedicated hardware designs are mandatory to reduce energy consumption in video systems. Although several solutions were proposed to previous standards, most of them cannot be directly used in AV1 and VVC, requiring a complete redesign.

This paper presents an extensive review of the complexity reduction solutions and dedicated hardware designs for AV1 and VVC published to date, and it is an invited extended version of our previous work [10].

II. AV1 AND VVC FEATURES
Both the AV1 and VVC encoders follow a hybrid block-based video coding scheme, which is based on the following signal and data processing operations, as presented in Fig. 1: (i) inter and intra-frame prediction, (ii) transform (T), (iii) quantization (Q), and (iv) entropy coding [11]. A reconstruction loop with inverse quantization (IQ) and inverse transform (IT) is also included to guarantee that the encoder and decoder will use the same reference data [11]. Finally,
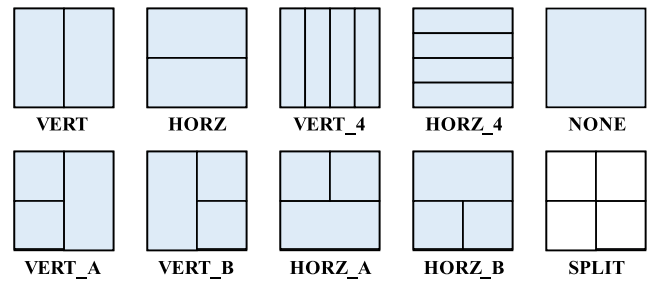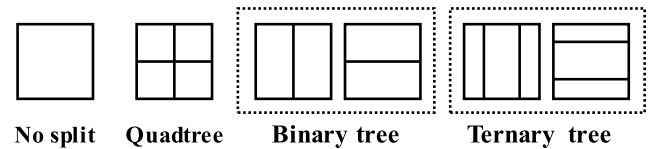
the in-loop filter is also used to increase coding efficiency and subjective image quality [11].

A. BLOCK PARTITIONING
Before the prediction process begins, a frame is divided into several blocks of pixels. Each video encoder defines a variable range of block sizes it can use.

In AV1, a video frame is partitioned in superblocks (SBs) of size 128×128 or 64×64 pixels. To deliver a locally optimal prediction for each SB, the encoder can further divide each SB using a 10-way partition tree structure, as illustrates Fig. 2. In the figure, filled partitions are final partition modes, but all four partitions of the unfilled partition (SPLIT) can be recursively divided based on the same 10-way tree structure, down to 4×4, which is the smallest supported block size [2].

In VVC, the video frame is partitioned in coding tree unities (CTUs) with 128×128 pixels. VVC inherits the Quadtree (QT) partitioning structure from HEVC and introduces the Multi-Type Tree (MTT) structure, resulting in the Quadtree with nested Multi-type Tree (QTMT). In VVC a CTU is first partitioned by a QT structure, then the quadtree structure can be further subdivided by an MTT structure [12]. As Fig. 3 illustrates, there are six partition types allowed in the QTMT structure, where a block, called coding unity (CU), can be defined as no split and the coding process is performed with the current CU size or the CU can be split using either the Quadtree, Binary tree, or Ternary tree structure [12]. The smallest allowed CU size is 4×4.

B. INTRA PREDICTION
In intra-frame prediction, each block is predicted from the reconstructed samples of previously encoded spatial neighbor blocks from the same frame. The VVC and AV1 intra predictions can be applied from 64×64 down to 4×4 block sizes and both formats allow the use of squared and rectangular block sizes.

AV1 supports 56 different directional predictors to explore spatial redundancies in directional textures. AV1 also supports a variety of non-directional predictors like (i) DC, similar to the one used in older codecs; (ii) Paeth, inspired in the VP9 True Motion mode; (iii) Smooth, Smooth Vertical, and Smooth Horizontal, which use a combined interpolation of horizontal and vertical neighbor samples; (iv) Recursive-based-filtering (RBF) modes 0 to 4 which allow an additional block partition; and (v) Chroma-from-luma (CFL), which reuses the luminance prediction to chrominance prediction [13]. There are also two modes particularly efficient for Screen Content Coding (SCC): Intra Block Copy [14] and Color Palette [15] modes.

VVC increases the number of angular prediction modes used in HEVC from 33 to 65. The planar and DC prediction modes follow the same approach employed in HEVC. VVC also introduces new intra coding tools, such as (i) Multiple Reference Line (MRL) [17], where three lines and columns are used in the prediction process; (ii) Intra Sub-Partition (ISP), where the CU can be further divided into sub-partitions [18]; (iii) Matrix-based Intra Prediction (MIP), where the matrix coefficients were defined by offline training using neural networks [19], [20]; (iv) Wide-angle prediction modes, for rectangular CUs [16]. and (iv) Cross-Component Linear Model (CCLM) [21], which was specially designed for chroma prediction to exploit cross-component redundancy.

C. INTER PREDICTION

In inter-frame prediction, the block is predicted from samples belonging to previously encoded frames. Both AV1 and VVC use Motion Estimation (ME) and Motion Compensation (MC) algorithms in addition to motion vector prediction tools to reduce the amount of lateral data. Both video formats allow block sizes from 128×128 to 4×4 in inter prediction. VVC and AV1 can evaluate 28 and 22 block sizes, respectively, in function of the difference between their frame partition processes.

This AV1 prediction may use up to seven reference frames [2]. Each one of these frames have a particular position and function, as presented in [22]. AV1 inter prediction brings a variety of novel solutions, like: (i) Warped Motion Compensation (WMC), using affine transformations to better capture non translational object movements [2]; (ii) Overlapped Block Motion Compensation (OBMC), which explores motion vectors from the neighborhood to improve the prediction quality around the borders [2]; (iii) Advanced Compound Prediction, where two references can be combined to form the prediction, with four main modes (compound wedge prediction, difference-modulated masked prediction, frame distance-based compound prediction and compound inter-intra prediction [2]); (iv) Dynamic Spatial and Temporal Motion Vector Referencing, with efficient motion vector prediction tools using candidate motion vectors from the spatial and temporal neighbors [2]. The accuracy of the AV1 motion vector is from 1/8 pixel. Sub-pixel accuracy is obtained by applying horizontal and vertical

interpolation filters over the pixels [2] and 90 different FIR filters are defined.

VVC enhances many of the inter prediction tools from HEVC, including the conventional ME and merge mode. The main VVC novelties are: (i) the use of Affine Motion Compensation (AMC) [24], to represent higher-order motion beyond translation, such as rotation, scaling, and shearing; (ii) the increase in the motion vector (MV) precision to 1/16 (compared to 1/4 in HEVC); (iii) Adaptive Motion Vector Resolution (AMVR), that allows the selection of the MV precision according to the coded mode [25]; (iv) Geometric Partitioning Mode (GPM) [26], which splits a CU into two non-rectangular partitions to perform the inter prediction separately; (v) Combined Inter and Intra Prediction (CIIP), which combines the inter prediction (merge mode) with the intra prediction (planar mode) [27]; (vi) Decoder-side MV refinement (DMVR), intending to improve the compression efficiency [28]; (vii) Bidirectional Optical Flow (BDOF), which explores the optical flow concept [29]; (viii) Prediction Refinement with Optical Flow (PROF), used for affine prediction also exploring the optical flow concept [4]; (ix) Bi-prediction with CU-level Weights (BCW), which uses till five predefined weights instead of the traditional weighted prediction [4]; (x) Extended Merge Prediction, allowing a new set of tools to improve the merge process [4].

D. TRANSFORMS AND QUANTIZATION

The prediction error, or the residues, between the intra and inter prediction and the original blocks are processed by the transform module (T module, in Fig. 1), which converts the values from the spatial domain to the frequency domain. Then, the quantization step (Q module in Fig. 1) is applied to the transformed coefficients to attenuate or eliminate values associated with spectral components that are not perceptually relevant for the human visual system.

AV1 specifies many square and non-square sizes for transform blocks, from 64×64 to 4×4. A rich set of transform kernels is defined for both inter and intra blocks consisting of 16 combinations of vertical and horizontal Discrete Cosine Transform (DCT), Asymmetrical Discrete Sine Transform (ADST), flipADST, and Identity Transform (IDTX) [30].

VVC introduces the Multiple Transform Selection (MTS) [31], Low-frequency Non-Separable Transform (LFNST) [32], and Sub-block Transform (SBT) [16] encoding tools for the transform module. MTS enhances the transform module by including Discrete Cosine Transform VIII (DCT-VIII) and Discrete Sine Transform VII (DST-VII) beyond DCT-II used as the main transform in the HEVC. Furthermore, non-square transforms are also supported in VVC and the supported block sizes range from 64×64 to 4×4. LFNST is a secondary transform applied to further decorrelate the coefficients obtained after the primary transformation for intra prediction. For inter prediction, VVC introduces the SBT, where only a subpart of the residual block is encoded.

TABLE 1. Summary results of AV1 complexity reduction solutions found in the literature.

Work	Version	Module	CQs	Configuration	BD-rate (%)	Complexity Reduction (%)
Jeong [41]	-	Intra	24, 33, 42, 51	kf-min-dist=1 kf-max-dist=1*	0.04	8.67
Jeong [42]	-	Intra	27, 36, 45, 54	kf-min-dist=1 kf-max-dist=1*	0.44	15.86
Gankhuyag [43]	-	Inter	24, 33, 42, 51	cpu-used=0, end-usage=cbr, pass=1, kf-min-dist=32, kf-max-dist=32, sb-size=64, threads=8, row-mt=1	9.38	16.37
Chiang [44]	-	Partitioning	-	-	0.61	64.14
Guo [45]	-	Partitioning	22, 27, 32, 37	-	1.04	36.80
Guo [46]	0.1.0	Partitioning / RDO	22**, 27, 32, 37, 42	cpu-used=0, end-usage=q, pass=1, kf-min-dist=0, kf-max-dist=9999, kf-mode=1, bit-depth=8, auto-altref=1, drop-frames=0	0.46	36.10
Kim [47]	1.0.0	Inter / Partitioning	32, 43, 55, 63	not informed	0.77	43.40
Chen [48]	1.0.0	Partitioning	27, 32, 37, 42	cpu-used=0, end-usage=q	0.79	37.80
Chen [49]	1.0.0	Filter	32, 43, 53, 63	enable-cdef=0, enable-restoration=0	4.10	4.12
Su [50]	1.0.0	Transform	20, 35, 50	cpu-used=0	0.12	21.81

*inferred from the experimental setup description, **not clear if authors use CQ 22 in the experiments

The quantization brings a small number of improvements in both encoders. In VVC, specifically, the Quantization Parameter (QP) was increased from 53 to 61, to allow higher compression rates, and the Dependent Quantization (DQ) tool was included allowing the use of two scalar quantizers [4].

E. ENTROPY CODING
The entropy coding processes the symbols (quantized coefficients and lateral data) to reduce their statistical redundancy by applying lossless algorithms.

AV1 uses a symbol-to-symbol adaptive multi-symbol arithmetic coder with the probability being updated every new symbol. Each syntax element in AV1 is a member of an alphabet of N elements, and a context consists of a set of N probabilities together with a small count to facilitate fast early adaptation [2].

As in HEVC, the Context-based Adaptive Binary Arithmetic Coder (CABAC) [33] is used in VVC. Some improvements were adopted in VVC, such as an engine with multi-hypothesis probability and an improved transform coefficient coding. For the probability estimation, the VVC maintains two estimators and computes the average probability between them for coding. Each estimator is independently updated with different adaptation rates, which are pre-trained based on the statistics of the associated bins. Both codecs exploit arithmetic coding at this step [34].

F. IN-LOOP FILTERING
The encoding process inserts artifacts in the decoded videos because of the block partitioning and the quantization level. Typical coding artifacts are blocking, ringing, and blurring. These artifacts decrease the video subjective quality and compromise the quality of prediction references. Thus, modern codecs use in-loop filtering to reduce these artifacts.

AV1 employs Deblocking Filter (DBF), Constrained Directional Enhancement Filter (CDEF) [35], and Loop Restoration Filters (LRF) [36], in this order. The 13-taps DBF is used to reduce the blocking effect that can be noticeable after coding. The CDEF is used to reduce ringing while preserving details. Finally, the LRF is composed of two filters, Wiener Filter [37] and Self-Guided Filter [38], to reduce blurring artifacts.

VVC uses four in-loop filters called Luma Mapping with Chroma Scaling (LMCS), Deblocking Filter (DF), Sample Adaptive Offset filter (SAO), and Adaptive Loop Filter (ALF) and Cross-Component Adaptive Loop Filter (CC-ALF) [16]. LMCS redistributes the amplitude of the input signal to better balance the dynamic range and improve the coding efficiency. The DF is used to reduce block artifacts. After, SAO is applied to attenuate ringing artifacts. Finally, ALF and CC-ALF reduce other potential distortions introduced by the transform and quantization process.

III. SYSTEM-LEVEL SOLUTIONS TARGETING AV1
This section discusses the main works in the literature focusing on system-level solutions targeting the AV1. These solutions contribute to solve a challenge related to the AV1 popularization: the extremely high computational effort required to process high-resolution videos. The works were divided into two main categories: (i) algorithmic optimizations intending to reduce the AV1 minimizing negative impacts in the coding efficiency, and (ii) dedicated hardware designs intending to reach real-time processing for high resolution videos, preferably considering power and energy constraints whereas minimizing negative impacts in the coding efficiency. Only one work cited in this section is based on a reference software version that precedes the AV1 released (libaom below 1.0.0), and whenever listed in tables, these works are sorted by libaom version.

A. ALGORITHMIC OPTIMIZATION FOR AV1
There are few papers published in the literature that deal with complexity reduction of the AV1 through algorithmic optimizations. Block partitioning optimization is the most published topic about complexity reduction on AV1, followed by inter and intra predictions. Table 1 presents a list of papers found in the literature that address complexity reduction through algorithmic optimizations, showing the Time Saving (TS) results and the coding efficiency of each solution measured using the BD-rate metric [40]. In this table,

the work [46] is the only one known to be based on a version that precedes the AV1 specification.

Complexity reduction solutions focusing on partitioning structures are presented in [44]–[48]. Chiang *et al.* [44] uses the rate-distortion cost (RD-cost) of the previous frame to infer the current partition tree depth limiting the partition types allowed to be predicted. A modification of RD-cost is also used by Guo *et al.* [46], which proposes a modification of the RD-cost process using a Bayesian model to provide an early termination for the partition tree process. The authors in [45] propose a method to find the partition structure in a particular case of use: the encoding of a video in multiple resolutions. The strategy derives the best superblock partitions for high-resolution videos after a statistical analysis in the co-located region of the low-resolution video.

In Kim *et al.* [47], the AV1 partition tree definition is early terminated during the prediction process in the inter prediction, also deciding if the compound inter prediction mode can be executed or not. Gankhuyag *et al.* [43] subdivides the UHD 8K 360 videos in tile boundaries limiting the inter prediction search area. Part of the high BD-rate in Gankhuyag *et al.* [43] can be explained because loop filters and some AV1 encoding options are disabled or restricted.

Considering the AV1 intra prediction, two works can be found in the literature. In Jeong *et al.* [41], the authors use the variation and Welford's online method with the RDO calculations made by the reference software to choose the best intra mode. In Jeong *et al.* [42], the authors propose the analysis of the intra prediction mode of the luma channel to include or not some intra prediction mode candidates for chroma channels, applied on HD or higher resolutions.

Complexity reduction for the transform coding stage is also proposed in Su *et al.* [50], where the authors propose to use a convolutional neural network (CNN) to decide the splitting process of the transform tree and to find the best transform modes candidates for the inter prediction blocks. Finally, Chen *et al.* [49] presents a CNN solution that replaces the AV1 in-loop filters. The solution presented by Chen *et al.* [49] presents modest results when applied exclusively on inter frames, but worse results are noticed when the solution is applied exclusively on the key frames, with a BD-rate higher than 10% with no time savings.

In summary, the most impressive results were found on papers focusing on reducing the partitioning process, with complexity reductions from 35% up to 64%. This type of optimization impacts all the encoding cycle since skipped block sizes are not evaluated by the encoder tools. Moreover, these works showed impacts of 1% or lower in BD-rate, even with the highest time saving results.

B. DEDICATED HARDWARE DESIGNS FOR AV1

Since the release of the AV1 bitstream specification, 11 hardware-related works have been published covering three areas: (i) intra-frame prediction, (ii) inter-frame prediction, and (iii) in-loop filtering.

Corrêa *et al.* [51]–[54] and Neto *et al.* [55] presented architectures for the intra prediction encoder, whereas Goebel *et al.* [56] presented an architecture for the decoder side. These architectures have in common the support to every possible block partition allowed by the AV1 specification. In [51] a non-directional intra prediction module for the encoder side, limited to a single prediction mode (Paeth) able to reach high throughput was presented. Massive parallelism is used to allow the processing of a whole 32×32 block (or any smaller block) in a single clock cycle. A throughput of UHD 8K at 30 frames per second (fps) was reported.

In [52] a non-directional intra prediction module for the encoder, limited to four intra prediction modes was presented. The authors decided to optimize all multiplication blocks to keep the area and power within feasible limits. The parallelism strategy allowed the processing of one block row/column per clock cycle, thus, the number of cycles depends on the width or height of the block, whichever is the largest. Similarly, in [53] a non-directional intra prediction module for the encoder capable of processing 11 non-directional modes was presented. A throughput of UHD 4K at 30 fps was reported for both architectures.

In [54] and [55], directional intra prediction designs for the encoder that share many similarities were described. Both designs support all 56 directional prediction modes, however, only the design proposed in [55] gives support to the four smoothing filtering processes of reference samples and the upsampling of reference samples. All 56 prediction modes are processed in parallel, one row/column per clock cycle. Although the number of prediction modes being processed in parallel is quite large, a significant amount of redundant operations is reused in [54], because all predicted blocks share the same reference samples. This, however, cannot be done with the same degree of efficiency in the architecture proposed in [55], because for each of the 56 prediction modes, the encoder can use a different configuration of smoothing filtering and upsampling of the reference samples. Thus, the reference samples are not shared among predicted blocks. A throughput of UHD 4K at 60 fps was reported for both architectures.

In [56] a non-directional intra prediction module for the decoder limited to the DC and CFL prediction modes was presented. The sample-level parallelism of the architecture allowed the processing of any block size as subblocks of size 4×4 (16 samples per cycle). In the decoding process, each block must be predicted only once using the prediction mode signaled in the bitstream, hence the CFL unit of this design is only used for CFL coded blocks, but the DC unit is used for both modes because the DC algorithm is one of the steps of the CFL prediction. The authors reported a throughput of UHD 4K at 60 fps.

Domanski *et al.* [57] and Freitas *et al.* [58] presented architectures for the subpixel interpolation filter present in the inter prediction module of the decoder. In [57], the sample-level parallelism of the architecture allows the processing of any block size as subblocks of size 4×4 (16 samples

TABLE 2. Summary results of AV1 hardware designs.

Work	Version	System	Module	Freq. (MHz)	Technology	Area (Kgates)	Power (mW)	Throughput
Freitas [58]	-	Decoder	Inter (FME)	282	STMicro 65nm	828.7	356.0	3840x2160@120fps
Corrêa [51]	1.0.0	Encoder	Intra	315	TSMC 40nm	247.3	268.4	7680x4320@30fps
Corrêa [52]	1.0.0	Encoder	Intra	648	TSMC 40nm	109.6	16.1	3840x2160@30fps
Corrêa [53]	1.0.0	Encoder	Intra	648	TSMC 40nm	128.5	65.6	3840x2160@30fps
Corrêa [54]	1.0.0	Encoder	Intra	1,296	TSMC 40nm	455.8	40.9	3840x2160@60fps
Neto [55]	1.0.0	Encoder	Intra	1,902	TSMC 40nm	691.7	382.1	3840x2160@60fps
Goebel [56]	1.0.0	Decoder	Intra	132	TSMC 40nm	89.4	7.96	3840x2160@60fps
Domanski [57]	1.0.0	Decoder	Inter (FME)	280	TSMC 40nm	141.1	81.3	7680x4320@30fps
Zummach [59]	1.0.0	Decoder	Filter	23	TSMC 40nm	369.9	65.1	3840x2160@60fps
Zummach [60]	1.0.0	Decoder	Filter	93.38	TSMC 40nm	185.36	43.29	3840x2160@60fps
Zummach [61]	1.0.0	Decoder	Filter	16.2	TSMC 40nm	39.35	3.96	3840x2160@60fps

per cycle), but since it is a decoder design, only one of the many supported filters is used per predicted block (the one signaled in the bitstream). The authors reported a throughput of UHD 8K at 30 fps. In [58], the level of parallelism is configurable, ranging from 4 to 128 samples per cycle. Because of this, the authors reported a throughput of UHD 8K at 120 fps, which is much higher than what was achieved by [57], but at the cost of a higher gate count and power dissipation.

Zummach *et al.* [59]–[61] presented architectures for the CDEF and DBF in-loop filters at the decoder side. In [59], a CDEF architecture for the decoder was presented. The CDEF process is applied to each area of size 8×8 within a frame, and the architecture was designed with enough parallelism to process an 8×8 area at every three clock cycles. The architecture is composed of a direction search unit, which classifies the input texture with one of eight directions, and a filtering core unit, which filters the input texture using 64 filter kernels based on the detected direction. In [60], another version of the architecture with lower parallelism was presented, this one capable of processing an area of 8×1 of the frame at every three cycles. The authors reported a throughput of UHD 4K at 60 fps for both designs. In [61], a DBF architecture for the decoder was presented. The architecture implements a parallelism of 56 samples per cycle, which is enough to allow a very low frequency when processing high resolution videos. The authors reported a throughput of UHD 4K at 60 fps.

Table 2 summarizes the results of the works discussed in this section. This table shows that all architectures are capable of real-time UHD processing. Encoder architectures required a high level of parallelism to achieve this throughput when compared to the decoder architectures. The reason for this is that the encoder must explore multiple possible ways of coding a single block, whereas the decoder must process the block according to what is signaled in the bitstream.

IV. SYSTEM-LEVEL SOLUTIONS TARGETING VVC
As in the previous section, this section discusses the main works in the literature focusing on system-level solutions, but now targeting the VVC. Again, the works were organized in two categories: (i) the works focusing on algorithmic optimizations, and (ii) the works presenting dedicated hardware designs. Every work mentioned in this section is based on

draft versions of the VVC specification (VTM version below 8.0), and whenever listed in tables, the works are sorted by VTM version.

A. ALGORITHMIC OPTIMIZATION FOR VVC
This section presents the related works focusing on complexity reduction solutions for VVC through algorithmic optimizations. Most of these works targeted the block partitioning structure of intra prediction employing statistical analysis and/or machine learning approaches. Some works also presented solutions to reduce the complexity in the evaluation of intra prediction modes, MTS step, and block partitioning of inter prediction.

Fu *et al.* [62] presented a fast CU partitioning algorithm using a classifier based on the Bayesian decision rule. The information derived from the current CU and horizontal binary splitting is used as a model input feature. This solution was evaluated with the All-Intra configuration, obtaining 45% of encoding time reduction and 1.02% of BD-rate increase.

Yang *et al.* [63] proposed a complexity reduction scheme composed of two strategies. The first one uses decision trees to avoid unnecessary block partition evaluations. In this case, a set of decision trees was trained for each partitioning type to decide whether to split the current block. The second one applies a gradient descent search to reduce the assessment of some angular intra prediction modes. The proposed solution can reduce 62.46% of the encoding time with a BD-rate increase of 1.93%, considering All-Intra configuration.

Chen *et al.* [64] developed a complexity reduction solution using the Support Vector Machine (SVM) to decide between horizontal and vertical partitioning. For this purpose, six SVM classifiers were considered according to the block size. The training of each classifier is carried out online during the encoding of the first frame, and the remaining frames are encoded considering the decision of the trained classifiers. This solution was evaluated with an All-Intra configuration. This solution reduces the encoding complexity by 50.97% with a BD-rate increase of 1.55%.

Lei *et al.* [65] developed a fast solution to avoid unnecessary block partition evaluations in advance, where a subset of directional intra-frame prediction modes are evaluated for virtual subpartitions of the current block to estimate the horizontal and vertical partitioning cost of the current

block. Based on the obtained costs, this solution can decide by avoiding horizontal or vertical partitions. The proposed solution was tested with the All-Intra configuration. The experimental results showed that this solution saves 40.7% of the encoding time with a 0.84% BD-rate increase.

Fu *et al.* [66] proposed an early decision scheme for the MTS evaluation in the intra-frame prediction. Based on the information from neighboring spatial blocks, the proposed scheme verifies if all neighbor blocks were encoded with DCT-II. If this is true, the evaluations of DST-VII and DCT-VIII are skipped; otherwise, the frequency of each transform type used in the neighboring and the transform list is ordered from most frequent transform to least frequent. In this case, if an intra prediction mode with the current transform obtained a higher cost than the previously evaluated transform, this prediction mode is discarded from the evaluations of the next transform types. This solution reduces 23% of the encoding complexity and increases the BD-rate by 0.16%, using All-Intra configuration.

Cui *et al.* [67] proposed a complexity reduction scheme based on the direction of the sample gradients to decide the best block partitioning structure. In this way, the scheme performs the decision on three partitioning possibilities, including split or not, horizontal or vertical, and BT or TT. The gradients of subpartitions in the current block are computed in four directions, including horizontal, vertical, 45°, and 135°, and compared with predefined threshold values. The proposed solution was evaluated with an All-Intra configuration and reduces 51.01% of the encoding complexity with an increase of 1.23% in the BD-rate.

Saldanha *et al.* [68] proposed a fast block partitioning decision scheme for deciding the direction of BT/TT partitions. The proposed scheme used the intra-frame prediction mode selected for the current block and the variance of subpartitions in the current block. The computation of the variance of subpartitions is performed to classify the texture direction as horizontal or vertical. If the variance classifies the block as horizontal texture and the selected angular intra prediction mode is horizontal, then the solution decides by skipping vertical partitions. This decision also is performed for skipping horizontal partitions. Besides, if the best mode in the current block is horizontal ISP, vertical partitions are skipped; otherwise, if the best mode is vertical ISP, the horizontal partitions are skipped. This scheme was evaluated using the All-Intra configuration, providing 31.41% of encoding time reduction with a 0.98% BD-rate increase.

Amestoy *et al.* [69] presented a solution based on Random Forest (RF) classifiers. For this purpose, a set of RF classifiers were defined to operate on three main decisions: (i) split or not, (ii) split with QT or BT/TT, (iii) split with horizontal or vertical partitioning. These classifiers consider the information of current block samples such as variance and horizontal and vertical gradients beyond encoding context information such as QP value and motion vectors. The proposed solution was evaluated with Random-Access

TABLE 3. Summary results of VVC complexity reduction solutions found in the literature.

Work	VTM version	Module	BD-rate (%)	Complexity Reduction (%)
Fu [62]	1.0	Partitioning	1.02	45.00
Yang [63]	2.0	Intra/ Partitioning	1.93	62.46
Chen [64]	2.1	Partitioning	1.55	50.97
Lei [65]	3.0	Partitioning	0.84	40.70
Fu [66]	3.0	Transform	0.16	23.00
Cui [67]	5.0	Partitioning	1.23	51.01
Saldanha [68]	5.0	Partitioning	0.98	31.41
Amestoy [69]	5.0	Partitioning (Inter)	0.61	30.10
Tissier [70]	6.1	Partitioning	0.75	42.20
Zhao [71]	7.0	Partitioning	0.86	39.39
Li [72]	7.0	Partitioning	1.32	44.65
Fan [73]	7.0	Partitioning	1.63	49.27

configuration, obtaining 30.1% of encoding complexity reduction with an increase of 0.61% in the BD-rate.

Tissier *et al.* [70], Zhao *et al.* [71], and Li *et al.* [72] proposed complexity reduction solutions based on Convolutional Neural Network (CNN) to choose the best block partitioning and to reduce the encoding complexity. The solution proposed in [70] was evaluated using All-Intra configuration. The experimental results showed a reduction of 42.2% in the encoding time with a 0.75% BD-rate increase. The proposed solutions in [71] and [72] were also evaluated using All-Intra configuration. The first one reduces the encoding time by 39.39%, with a BD-rate increase of 0.86%. The second one provides 44.65% of encoding time reduction with 1.32% BD-rate increase.

Fan *et al.* [73] developed a solution based on the current block variance, subpartition variance, and Sobel filter. The current block variance is employed to verify the homogeneity of 32×32 blocks and early terminate the QTMT evaluation. The variance of subpartitions is calculated to choose only one partition type among QT, BTH, BTV, TTH, and TTV. The Sobel filter is used to decide early by the QT partitioning and skip the MTT structure evaluation. The proposed solution was evaluated under All-Intra configuration. The experimental results demonstrated that the proposed solution increases 1.63% in BD-rate and reduces 49.27% of the encoding time.

Table 3 summarizes the works presented in this section. These works focused on QTMT, intra prediction modes, and MTS coding tools. One can notice that most of these works targeted the VVC block partitioning mainly for intra-frame prediction since its process has the computational burden significantly increased compared to the predecessor standards. Additionally, one work focuses on accelerating the QTMT decision considering the inter-frame prediction, and another work targets at the MTS transforms step. Seven works developed solutions based on machine learning, such as Bayesian theorem, decision trees, SVM, and CNN, and five works proposed heuristics based on statistical analysis. These works used encoder context information and/or statistical

information of block samples to build efficient complexity reduction solutions capable of reducing the number of encoding modes evaluated.

It is important to highlight that the works listed in Table 3 are based on early versions of the VTM reference software (i.e., before the VVC standardization). However, the proposed approaches can be used directly or with minor modifications in newer VTM versions to provide similar performance, especially for the block partitioning structure that remains without significant changes since the first VTM version. For instance, the approach used in the work [68] provides 31% of complexity reduction with a 0.99% BD-rate increase considering VTM 5.0, and directly implemented in the VTM 10.0 it reaches 30% of complexity reduction with 0.92% of BDBR increase.

B. DEDICATED HARDWARE DESIGNS FOR VVC

Although there are few works in the literature related to hardware architecture designs for the different modules of VVC, in this section we discuss the main published works. These works focused on intra-frame prediction, fractional interpolation filters, and transform, considering FPGA and ASIC-based designs.

Azgin *et al.* [74] designed a hardware architecture for the VVC intra prediction encoder. This architecture encompasses only the 65 angular prediction modes and square-shaped blocks with sizes ranging from 4×4 to 32×32. In this work, a strategy of data reuse was employed to calculate identical prediction equations only once. This strategy explores identical prediction equations that are used in angular prediction modes for the same or different block sizes. Besides, the equations are computed by using DPS blocks aiming for low-energy consumption and faster multiplication results. The proposed architecture was synthesized for a Xilinx Virtex-7 FPGA, being capable of processing 1920×1080 videos at 34 fps.

Mert *et al.* [75] and Azgin *et al.* [76], [77] proposed hardware architectures for VVC fractional interpolation filters. The works [75] and [76] considered the original fractional interpolation filters without any modification. In [77], an optimized version of these implementations was designed. This solution considers that neighboring samples have similar values due to the spatial correlation, then small coefficient values have little influence on the filter results. Based on this idea, this optimized version implements an approximation of the original VVC filter coefficients, where 14 different 3-tap filters and one 4-tap filter were designed in place of 15 8-tap filters defined in the VVC. The proposed approximate VVC fractional filters were implemented in the VTM and presented a BDBR of 0.52%. The synthesis results were obtained for a Xilinx Virtex-7 FPGA. The authors reported a throughput of 1920×1080 videos at 47 fps. Additionally, a comparison with the baseline implementation of VVC (15 8-tap filters) demonstrated that the proposed hardware design reduces the power dissipation up to 40%.

The works [78]–[85] present hardware architectures for the VVC transform module considering the encoder and decoder systems. In these works, different hardware technologies and different approaches were considered to implement the forward and inverse transforms. From these papers, we selected to discuss the works [83] and [85], since one solution targets the decoder and other targets the encoder and both targeted ASIC technologies and used the newest reference software versions among the related works,

Fan *et al.* [83] proposed a high-performance pipelined hardware implementation for 2D DST-VII/DCT-VIII transform operations in VVC. The developed hardware architecture considers square and rectangular-shaped block sizes, except 64×64 block size. The proposed solution consists of three main modules: (i) 1D row transform, (ii) 1D column transform, and (iii) transpose memory. The modules (i) and (ii) have the same structure, only with different bit-width parameters. The 1D transform modules use Shift-Addition Units (SAUs) to perform the matrix multiplications, and to reduce the number of operations the SAUs were designed by employing the N-Dimensional Reduced Adder Graph (RAG-n). Besides, both the DST-VII and DCT-VIII have the same coefficients in each row but in inverse order, consequently, the hardware design takes advantage of DST-VII architecture for the implementation of DCT-VIII with no additional complexity by only inverting the order of the inputs and assigning adequate outputs signs. A dual-port SRAM is used to implement the transpose memory since the pipelined hardware implementation needs to read and write simultaneously. The proposed solution considers that all block sizes are within a 32×32 block size. The synthesis results were generated targeting an ASIC TSMC 65 nm at 250 MHz. The hardware design can process 7680×4320 videos at 160 fps with a total area of 496.4Kgates and power dissipation of 62.6 mW. Besides, the authors compared the proposed solution with an approach using multipliers and register array, where the proposed solution reduces the area and power by 47% and 10%, respectively.

Farhat *et al.* [85] developed a hardware architecture for the transform module of the VVC decoder. The hardware architecture supports 1D Inverse DCT-II (IDCT-II) and IDST-VII/IDCT-VIII with sizes ranging from 4 to 64 and from 4 to 32, respectively. The hardware design has a fixed throughput of two samples per cycle and a fixed latency for all block sizes based on the largest transform block latency to provide a fully pipelined structure. The authors proposed two designs: (i) using adders and shifts instead of multipliers (MCM) and (ii) using regular multipliers (RM). The synthesis results were generated targeting an ASIC TSMC 28 nm at 600 MHz. The authors reported that the architecture using RM consumed 63% fewer gates than MCM, being capable of processing 3840×1260 videos at 30 fps.

Table 4 summarizes the works discussed in this section. These works focused on the novel VVC tools such as the

CORRÊA *et al.*: AV1 AND VVC VIDEO CODECS: OVERVIEW ON COMPLEXITY REDUCTION AND HARDWARE DESIGN

TABLE 4. Summary results of VVC hardware designs.

Work	Version	System	Module	Freq. (MHz)	Technology	Area (Kgates)	Power (mW)	Throughput
Azgin [74]	-	Encoder	Intra	119	FPGA 28nm	46.4 KLUTs	-	1920x1080@34fps
Mert [75]	-	Encoder	Inter (FME)	435	ASIC 90nm	37.6 Kgates	467.9	1920x1080@88fps
Azgin [76]	-	Encoder	Inter (FME)	357	ASIC 90nm	11.7 Kgates	77.1	3840x2160@95fps
Azgin [77]		Encoder	Inter (FME)	227	FPGA 28nm	9.3 KLUTs	320.18	1920x1080@47fps
Garrido [78]	-	Encoder	Transform	577	FPGA 14nm	1.6 KALMs	-	3840x2160@86fps
Kammoun [79]	BMS	Enc/Dec	Transform	257	FPGA 20nm	45.4 KALMs	-	3840x2160@94fps
Kammoun [80]	JEM 4.0	Encoder	Transform	147	FPGA 20nm	113.0 KALMs	-	1920x1080@34fps
Kammoun [81]	VTM 3.0	Enc/Dec	Transform	228	FPGA 20nm	36.8 KALMs	-	3840x2160@96fps
Yibo [82]	VTM 4.0	Encoder	Transform	384	ASIC 65nm	228.6 Kgates	127.36	-
Fan [83]	VTM 4.0	Encoder	Transform	250	ASIC 65nm	496.4 Kgates	62.6	7680x4320@160fps
Garrido [84]	VTM 4.2	Encoder	Transform	200	FPGA 28nm	5.1 KALMs	-	1920x1080@40fps
Farhat [85]	VTM 6.0	Decoder	Transform	600	ASIC 28nm	96.9 Kgates	-	3840x2160@30fps

new intra prediction modes, the new fractional interpolation filters, and the new transform set. Besides, the proposed solutions explored advanced techniques of hardware design, such as data reuse, operator optimization, and pipeline, to provide high throughput (at least 1,920×1,080 at 30 fps) with low hardware resource usage and low power dissipation.

The software versions Joint Exploration Model (JEM) and Benchmark Set (BMS) in Table 4 were created for the testing of proposed tools to achieve coding efficiency beyond HEVC, and both were discontinued after the release of VTM. Once again, although these works are based on early versions of the reference software, the proposed approaches could be adjusted to comply to the current standard.

V. OPEN RESEARCH TOPICS

The number of published works focusing on algorithmic optimizations targeting the AV1 and VVC complexity reduction is still small. Most works discussed in Section III-A focus on the prediction and partitioning problems, which are well known to be bottlenecks of the encoding cycle, whereas the ones presented in Section IV-A focus heavily on optimizing the partitioning algorithm. This trend is justified by the undeniable fact that optimizations on both prediction and partitioning algorithms will result in less prediction candidates being evaluated by the RDO, which will lead to faster encoding times and less intensive memory access.

The hardware designs for both the AV1 and VVC, presented respectively on Sections III-B and IV-B, are mainly about accelerating the prediction modules and transforms through efficient parallel designs, although filter designs are also discussed. Some works are concerned about reaching real-time processing and low energy consumption on the compute-intensive encoder side, others are concerned about achieving low area and low power decoder designs. The trend on transforms related designs is due to the considerable number of operations required by transform algorithms, on both the encoder and decoder.

When compared to years of research on consolidated standards, such as the H.264 and HEVC, we conclude that the research on the current generation formats still does not cover all parts of the video coding cycle, despite the

fact every field has space for novel algorithms and hardware designs to allow more efficient encoding and decoding. Moreover, when considering all modules of the codec working together, more problems come to surface, notably: (i) the very high complexity of the RDO decision in the encoder, (ii) the large number of memory accesses, which makes parallelism exploration and low energy consumption a big challenge, and (iii) the need for low energy designs, despite the complexity of the algorithms and the required memory accesses.

Some of these challenges are being efficiently faced using machine learning and deep learning approaches and further investigations using these approaches tend to generate good results. Other challenges are being faced through dedicated hardware designs, mainly targeting high throughput with low energy consumption. But in both cases, novel contributions are still needed and further research are mandatory. This conclusion is supported when analyzing the works presented in this paper, where is notably that the presented solutions are not able to completely handle the challenges of this area.

In one side, some works targeting algorithmic optimizations, even presenting important results, are also only focusing on specific encoder modules and global encoder optimizations are not presented in the literature. Besides, even with expressive encoder complexity reductions, the residual encoder complexity is still very high, avoiding efficient implementations mainly for a scenario focusing on battery powered devices.

In other side, there is an important lack of hardware solutions for the AV1 and VVC available in the literature, and although designs for specific modules exist, a complete encoder or decoder has not been proposed yet

A promising approach to face this challenge is the join exploration of advanced hardware designs techniques together with video codec algorithmic optimizations. This is an open research topic that can bring important. Considering the hardware designs, approximate computing (in many levels) should be a trend for efficient designs. Other traditional techniques also must be explored, like: (i) dedicated memory hierarchy, (ii) pipeline, (iii) multiplierless solutions, (iv) data, clock, and power gating, among others. Considering the algorithmic level, machine learning and deep learning, as cited

VOLUME 2, 2021

before, seems to be inevitable to allow expressive complexity reductions.

It is important to highlight that the algorithmic optimizations must be done considering its future hardware implementation. This means that the algorithmic optimization cannot just focus on execution time reduction when running the encoder or decoder in a general-purpose processor, but must also consider questions like: (i) avoid data dependencies, to allow the parallelism exploration in hardware; (ii) reduce memory bandwidth, since this is the main hardware bottleneck; (iii) avoid the use of float point operations and multipliers and dividers when using integer operations, among others.

Finally, the video transcoding task is another example of research topic that needs more attention, and although it is possible to find papers about AV1 or VVC transcoding, there is no hardware proposal to deal with this problem yet.

VI. CONCLUSION

This article presented an extensive review of published works related to algorithmic optimizations and dedicated hardware designs for the state-of-the-art AV1 and VVC video coding formats. These topics are of major relevance to the Circuits and Systems (CAS) community since next-generation video encoding and decoding applications will demand dedicated and optimized hardware design accompanied by efficient complexity reduction algorithms.

Regarding complexity reduction works, we identified that most works focus on optimizing the encoder decision on block partition and decision of prediction modes. Considering the hardware-related works, the focus was to achieve real-time when processing high resolutions, and to do that using as low energy as possible. The works were focused in the AV1 or VVC modules that demand the higher number of operations, but complete encoder or decoder implementations were not reported.

Therefore, our conclusion is that there is still important open research topics and challenges related to AV1 and VVC encoders and decoders. The most promising solution is to explore algorithmic optimizations together with advanced hardware design techniques, joining the optimization efforts to reach high-throughput and low-energy solutions able to process ultra-high-definition videos in battery powered devices.

ACKNOWLEDGMENT

REFERENCES

[1] Cisco Systems. (2018). *Global 2022 Forecast Highlights*. Accessed: Mar. 2021. [Online]. Available: https://www.cisco.com/c/dam/m/en_us/solutions/service-provider/vni-forecast-highlights/pdf/Global_2022_Forecast_Highlights.pdf

[2] J. Han *et al.*, "A technical overview of AV1," 2021. [Online]. Available: https://arxiv.org/abs/2008.06091.

[3] P. Rivaz and J. Haughton, *AV1 Bitstream & Decoding Process Specification*, Alliance Open Media, Wakefield, MA, USA, 2019. Accessed: Mar. 2021. [Online]. Available: https://aomedia.org/av1/specification/

[4] B. Bross *et al.*, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Aug. 2, 2021, doi: 10.1109/TCSVT.2021.3101953.

[5] *Information Technology: Coded Representation of Immersive Media—Part 3: Versatile Video Coding*, Standard ISO/IEC DIS 23090-3, 2020.

[6] D. Mukherjee *et al.*, "A technical overview of VP9—The latest open-source video codec," *SMPTE Motion Imag. J.*, vol. 124, no. 1, pp. 44–54, Jan. 2015.

[7] *Information Technology: High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, Standard ISO/IEC 23008-2, 2013.

[8] *Information Technology: Coding of Audio-Visual Objects—Part 10: Advanced Video Coding*, Standard ISO/IEC 14496-10, 2003.

[9] *Information Technology: Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video*, Standard ISO/IEC 13818-2, 1996.

[10] M. Saldanha *et al.*, "An overview of dedicated hardware designs for state-of-the-art AV1 and H.266/VVC video codecs," in *Proc. 27th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, Glasgow, U.K., 2020, pp. 1–4.

[11] G. Corrêa, P. Assunção, L. Agostini, and L. A. da Silva Cruz, *Complexity-Aware High Efficiency Video Coding*. Cham, Switzerland: Springer, 2016. p. 225, doi: 10.1007/978-3-319-25778-5.

[12] Y.-W. Huang *et al.*, "A VVC proposal with quaternary tree plus binary-ternary tree coding block structure and advanced coding techniques," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 5, pp. 1311–1325, May 2020.

[13] L. Trudeau, N. Egge, and D. Barr, "Predicting chroma from Luma in AV1," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, 2018, pp. 374–382.

[14] J. Li *et al.*, "Intra block copy for screen content in the emerging AV1 video codec," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, 2018, pp. 355–364.

[15] L. Guo, W. Pu, F. Zou, J. Sole, M. Karczewicz, and R. Joshi, "Color palette for screen content coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Paris, France, 2014, pp. 5556–5560.

[16] J. Chen, Y. Ye, and S. Kim, *Algorithm description for Versatile Video Coding and Test Model 10 (VTM 10)*, document JVET 19th Meeting, JVET-S2002, JVET, Geneva Switzerland, Jul. 2019.

[17] B. Bross *et al.*, *CE3: Multiple Reference Line Intra Prediction (Test 1.1.1, 1.1.2, 1.1.3 and 1.1.4)*, JVET 12th Meeting, JVET-L0283, JVET, Macau, China, Oct. 2018.

[18] S. De-Luxán-Hernández *et al.*, "An intra subpartition coding mode for VVC," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, 2019, pp. 1203–1207.

[19] M. Schäfer *et al.*, "An affine-linear intra prediction with complexity constraints," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, 2019, pp. 1089–1093.

[20] J. Pfaff *et al.*, "Data-driven intra-prediction modes in the development of the versatile video coding standard," *ITU J. ICT Discoveries*, vol. 3, no. 1, 2020. [Online]. Available: http://handle.itu.int/11.1002/pub/8153d787-en

[21] K. Zhang, J. Chen, L. Zhang, X. Li, and M. Karczewicz, "Multi-model based cross-component linear model chroma intra-prediction for video coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St. Petersburg, FL, USA, 2017, pp. 1–4.

[22] W.-T. Lin *et al.*, "Efficient AV1 video coding using a multi-layer framework," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, 2018, pp. 365–373.

[23] W.-J. Chien, Y. Chen, J. Chen, L. Zhang, M. Karczewicz, and X. Li, "Sub-block motion derivation for merge mode in HEVC," in *Proc. SPIE Appl. Digit. Image Process. XXXIX*, vol. 9971, 2016, Art. no. 99711K, doi: 10.1117/12.2239709.

[24] K. Zhang, Y.-W. Chen, L. Zhang, W.-J. Chien, and M. Karczewicz, "An improved framework of affine motion compensation in video coding," *IEEE Trans. Image Process.*, vol. 28, pp. 1456–1469, 2019.

[25] Z. Wang, J. Zhang, N. Zhang, and S. Ma, "Adaptive motion vector resolution scheme for enhanced video coding," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, USA, 2016, pp. 101–110.

[26] H. Gao, S. Esenlik, E. Alshina, and E. Steinbach, "Geometric partitioning mode in versatile video coding: Algorithm review and analysis," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Nov. 24, 2020, doi: 10.1109/TCSVT.2020.3040291.

[27] C. Chen, X. Xiu, Y. He, and Y. Ye, "Generalized bi-prediction method for future video coding," in *Proc. Picture Coding Symp. (PCS)*, Nuremberg, Germany, 2016, pp. 1–5.

[28] H. Gao, H. Gao, X. Chen, S. Esenlik, J. Chen, and E. Steinbach, "Decoder-side motion vector refinement in VVC: Algorithm and hardware implementation considerations," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3197–3211, Aug. 2021.

[29] A. Alshin, E. Alshina, and T. Lee, "Bi-directional optical flow for improving motion compensation," in *Proc. 28th Picture Coding Symp.*, Nagoya, Japan, 2010, pp. 422–425, doi: 10.1109/PCS.2010.5702525.

[30] S. Parker *et al.*, "On transform coding tools under development for VP10," in *Proc. SPIE Appl. Digit. Image Process. XXXIX*, vol. 9971, 2016, Art. no. 997119, doi: 10.1117/12.2239105.

[31] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W.-J. Chien, "Enhanced multiple transform for video coding," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, USA, 2016, pp. 73–82.

[32] M. Koo, M. Salehifar, J. Lim, and S.-H. Kim, "Low frequency non-separable transform (LFNST)," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, 2019, pp. 1–5.

[33] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[34] H. Schwarz *et al.*, "Quantization and entropy coding in the versatile video coding (VVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Apr. 9, 2021, doi: 10.1109/TCSVT.2021.3072202.

[35] S. Midtskogen and J.-M. Valin, "The Av1 constrained directional enhancement filter (Cdef)," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Calgary, AB, Canada, 2018, pp. 1193–1197.

[36] D. Mukherjee, S. Li, Y. Chen, A. Anis, S. Parker, and J. Bankoski, "A switchable loop-restoration with side-information framework for the emerging AV1 video codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Beijing, China, 2017, pp. 265–269.

[37] M. Siekmann, S. Bosse, H. Schwarz, and T. Wiegand, "Separable Wiener filter based adaptive in-loop filter for video coding," in *Proc. 28th Picture Coding Symp.*, Nagoya, Japan, 2010, pp. 70–73.

[38] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013, doi: 10.1109/TPAMI.2012.213.

[39] Alliance for Open Media. (2021). *AOM Git at Google*. Accessed: Mar. 2021. [Online]. Available: https://aomedia.googlesource.com/aom

[40] G. Bjøntegaard. (2001). *Calculation of Average PSNR Differences Between RD-Curves*. Accessed: Mar. 2021. [Online]. Available: https://www.itu.int/wftp3/av-arch/video-site/0104_Aus/VCEG-M33.doc

[41] J. Jeong, G. Gankhuyag, and Y.-H. Kim, "A fast intra mode decision based on accuracy of rate distortion model for AV1 intra encoding," in *Proc. 34th Int. Techn. Conf. Circuits Syst. Comput. Commun. (ITC-CSCC)*, JeJu, South Korea, 2019, pp. 1–3.

[42] J. Jeong, G. Gankhuyag, and Y.-H. Kim, "Fast chroma prediction mode decision based on luma prediction mode for AV1 intra coding," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, 2019, pp. 1050–1052.

[43] G. Gankhuyag, J. Jeong, and Y.-H. Kim, "Advanced motion-constrained AV1 encoder for 8K 360 VR tiled streaming," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Jeju, South Korea, 2019, pp. 682–684.

[44] C.-H. Chiang, J. Han, and Y. Xu, "A multi-pass coding mode search framework for AV1 encoder optimization," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, USA, 2019, pp. 458–467.

[45] B. Guo, Y. Han, and J. Wen, "Fast block structure determination in Av1-based multiple resolutions video encoding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, San Diego, CA, USA, 2018, pp. 1–6.

[46] B. Guo, X. Chen, J. Gu, Y. Han, and J. Wen, "A Bayesian approach to block structure inference in AV1-based multi-rate video encoding," in *Proc. Data Compression Conf.*, Snowbird, UT, USA, 2018, pp. 383–392.

[47] J. Kim, S. Blasi, A. S. Dias, M. Mrak, and E. Izquierdo, "Fast inter-prediction based on decision trees for AV1 encoding," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Brighton, U.K., 2019, pp. 1627–1631.

[48] X. Chen, B. Guo, M. Tang, Y. Han, and J. Wen, "A conditional Bayesian block structure inference model for optimized AV1 encoding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, 2019, pp. 1270–1275.

[49] G. Chen, D. Ding, D. Mukherjee, U. Joshi, and Y. Chen, "AV1 in-loop filtering using a wide-activation structured residual network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, 2019, pp. 1725–1729.

[50] H. Su, M. Chen, A. Bokov, D. Mukherjee, Y. Wang, and Y. Chen, "Machine learning accelerated transform search for AV1," in *Proc. Picture Coding Symp. (PCS)*, Ningbo, China, 2019, pp. 1–5.

[51] M. Corrêa, B. Waskow, J. Goebel, D. Palomino, G. Corrêa, and L. Agostini, "A high throughput hardware architecture targeting the AV1 Paeth intra predictor," in *Proc. IEEE 10th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Armenia, Colombia, 2019, pp. 93–96.

[52] M. Corrêa, B. Waskow, B. Zatt, D. Palomino, G. Corrêa, and L. Agostini, "High throughput hardware design for AV1 Paeth and smooth intra modes," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, 2019, pp. 1–5.

[53] M. M. Corrêa, B. H. Waskow, J. W. Goebel, D. M. Palomino, G. R. Corrêa, and L. V. Agostini, "A high-throughput hardware architecture for AV1 non-directional intra modes," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 5, pp. 1481–1494, May 2020.

[54] M. Corrêa, L. Neto, D. Palomino, G. Corrêa, and L. Agostini, "ASIC solution for the directional intra prediction of the AV1 encoder targeting UHD 4K videos," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, 2020, pp. 1–5.

[55] L. Neto, M. Corrêa, D. Palomino, L. Agostini, and G. Correa, "Directional intra frame prediction architecture with edge filter and upsampling for AV1 video coding," in *Proc. 33rd Symp. Integr. Circuits Syst. Design (SBCCI)*, Campinas, Brazil, 2020, pp. 1–6.

[56] J. Goebel, B. Zatt, L. Agostini, and M. Porto, "Hardware design of DC/CFL intra-prediction decoder for the AV1 codec," in *Proc. 32nd Symp. Integr. Circuits Syst. Design (SBCCI)*, Sao Paulo, Brazil, 2019, pp. 1–6.

[57] R. Domanski *et al.*, "High-throughput multifilter interpolation architecture for AV1 motion Compensation," *IEEE Trans. Circuits Syst. II Exp. Briefs*, vol. 66, no. 5, pp. 883–887, May 2019.

[58] D. Freitas, R. da Silva, Í. Siqueira, C. M. Diniz, R. A. L. Reis, and M. Grellert, "Hardware architecture for the regular interpolation filter of the AV1 video coding standard," in *Proc. 28th Eur. Signal Process. Conf. (EUSIPCO)*, Amsterdam, The Netherlands, 2021, pp. 560–564.

[59] E. Zummach, R. Palau, J. Goebel, L. Agostini, and M. Porto, "High-throughput CDEF architecture for the AV1 decoder targeting 4K@60fps videos," in *Proc. IEEE 11th Latin Amer. Symp. Circuits Syst. (LASCAS)*, San Jose, Costa Rica, 2020, pp. 1–4.

[60] E. Zummach, R. Palau, J. Goebel, D. Palomino, L. Agostini, and M. Porto, "Efficient hardware design for the AV1 CDEF filter targeting 4K UHD videos," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, 2020, pp. 1–5.

[61] E. Zummach *et al.*, "An UHD 4K@60fps deblocking filter hardware targeting the AV1 decoder," in *Proc. 27th IEEE Int. Conf. Electron. Circuits Syst. (ICECS)*, Glasgow, U.K., 2020, pp. 1–4.

[62] T. Fu, H. Zhang, F. Mu, and H. Chen, "Fast CU partitioning algorithm for H.266/VVC intra-frame coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, 2019, pp. 55–60.

[63] H. Yang, L. Shen, X. Dong, Q. Ding, P. An, and G. Jiang, "Low-complexity CTU partition structure decision and fast intra mode decision for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1668–1682, Jun. 2020.

[64] F. Chen, Y. Ren, Z. Peng, G. Jiang, and X. Cui, "A fast CU size decision algorithm for VVC intra prediction based on support vector machine," *Multimedia Tools Appl.*, vol. 79, pp. 27923–27939, Jul. 2020.

[65] M. Lei, F. Luo, X. Zhang, S. Wang, and S. Ma, "Look-ahead prediction based coding unit size pruning for VVC intra coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, 2019, pp. 4120–4124.

[66] T. Fu, H. Zhang, F. Mu, and H. Chen, "Two-stage fast multiple transform selection algorithm for VVC intra coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shanghai, China, 2019, pp. 61–66.

[67] J. Cui, T. Zhang, C. Gu, X. Zhang, and S. Ma, "Gradient-based early termination of CU partition in VVC intra coding," in *Proc. Data Compression Conf. (DCC)*, Snowbird, UT, USA, 2020, pp. 103–112.

[68] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Fast partitioning decision scheme for versatile video coding intra-frame prediction," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Seville, Spain, 2020, pp. 1–5.

[69] T. Amestoy, A. Mercat, W. Hamidouche, D. Menard, and C. Bergeron, "Tunable VVC frame partitioning based on lightweight machine learning," *IEEE Trans. Image Process.*, vol. 29, pp. 1313–1328, 2020.

[70] A. Tissier, W. Hamidouche, J. Vanne, F. Galpin, and D. Menard, "CNN oriented complexity reduction of VVC intra encoder," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, UAE, 2020, pp. 3139–3143.

[71] J. Zhao, Y. Wang, and Q. Zhang, "Adaptive CU split decision based on deep learning and multifeature fusion for H.266/VVC," *Sci. Program.*, vol. 2020, Aug. 2020, Art. no. 8883214. doi: 10.1155/2020/8883214.

[72] T. Li, M. Xu, and R. Tang, "DeepQTMT: A deep learning approach for fast QTMT-based CU partition of intra-mode VVC," 2020. [Online]. Available: https://arxiv.org/abs/2006.13125.

[73] Y. Fan, J. Chen, H. Sun, J. Katto, and M. Jing, "A fast QTMT partition decision strategy for VVC intra prediction," *IEEE Access*, vol. 8, pp. 107900–107911, 2020.

[74] H. Azgin, E. Kalali, and I. Hamzaoglu, "An efficient FPGA implementation of versatile video coding intra prediction," in *Proc. 22nd Euromicro Conf. Digit. Syst. Design (DSD)*, Kallithea, Greece, 2019, pp. 194–199.

[75] A. C. Mert, E. Kalali, and I. Hamzaoglu, "A low power versatile video coding (VVC) fractional interpolation hardware," in *Proc. Conf. Design Archit. Signal Image Process. (DASIP)*, Porto, Portugal, 2018, pp. 43–47.

[76] H. Azgin, A. C. Mert, E. Kalali, and I. Hamzaoglu, "A reconfigurable fractional interpolation hardware for VVC motion compensation," in *Proc. 21st Euromicro Conf. Digit. Syst. Design (DSD)*, Prague, Czech Republic, 2018, pp. 99–103.

[77] H. Azgin, E. Kalali, and I. Hamzaoglu, "An approximate versatile video coding fractional interpolation hardware," in *Proc. IEEE Int. Conf. Consum. Electron. (ICCE)*, Las Vegas, NV, USA, 2020, pp. 1–4.

[78] M. J. Garrido, F. Pescador, M. Chavarrías, P. J. Lobo, and C. Sanz, "A 2-D multiple transform processor for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 65, no. 3, pp. 274–283, Aug. 2019.

[79] A. Kammoun, W. Hamidouche, F. Belghith, J.-F. Nezan, and N. Masmoudi, "Hardware design and implementation of adaptive multiple transforms for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 424–432, Nov. 2018.

[80] A. Kammoun, W. Hamidouche, P. Philipp, F. Belghith, N. Massmoudi, and J.-F. Nezan, "Hardware acceleration of approximate transform module for the versatile video coding standard," in *Proc. 27th Eur. Signal Process. Conf. (EUSIPCO)*, A Coruna, Spain, 2019, pp. 1–5.

[81] A. Kammoun *et al.*, "Forward-inverse 2D hardware implementation of approximate transform core for the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4340–4354, Nov. 2020.

[82] F. Yibo, K. Jiro, S. Heming, Z. Xiaoyang, and Z. Yixuan, "A minimal adder-oriented 1D DST-VII/DCT-VIII hardware implementation for VVC standard," in *Proc. 32nd IEEE Int. Syst. Chip Conf. (SOCC)*, Singapore, 2019, pp. 176–180.

[83] Y. Fan, Y. Zeng, H. Sun, J. Katto, and X. Zeng, "A pipelined 2D transform architecture supporting mixed block sizes for the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3289–3295, Sep. 2020.

[84] M. J. Garrido, F. Pescador, M. Chavarrías, P. J. Lobo, C. Sanz, and P. Paz, "An FPGA-based architecture for the versatile video coding multiple transform selection core," *IEEE Access*, vol. 8, pp. 81887–81903, 2020.

[85] I. Farhat, W. Hamidouche, A. Grill, D. Menard, and O. Déforges, "Lightweight hardware implementation of VVC transform block for ASIC decoder," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, 2020, pp. 1663–1667.

MARCEL CORRÊA received the B.S. and M.S. degrees in computer science from the Federal University of Pelotas (UFPel), Pelotas, Brazil, in 2013 and 2017, respectively, where he is currently pursuing the Ph.D. degree. He is a Professor with the Sul-rio-grandense Federal Institute of Science, Education and Technology (IFSul), Brazil. He is also a contributing member of UFPel's Video Technology Research Group and has been with the Group of Architectures and Integrated Circuits since 2009. His main topics of interest are video coding, data compression, and hardware design.

MÁRIO SALDANHA received the B.S. and M.Sc. degrees in computer science from the Federal University of Pelotas (UFPel), Pelotas, RS, Brazil, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in computer science. He is a member of the Video Technology Research Group, UFPel. His research interests include complexity reduction and hardware-friendly algorithms for 2-D/3-D video coding.

ALEX BORGES (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the Federal University of Pelotas, Brazil, in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree in computer science. He has been a Researcher with the Video Technology Research Group since 2015. His research interests include algorithms for video compression, video coding standards, and video transcoding.

GUILHERME CORRÊA (Senior Member, IEEE) received the M.S. degree in computer science from the Federal University of Rio Grande do Sul (UFRGS) in 2010, and the Ph.D. degree in electrical and computer engineering from the University of Coimbra, Portugal, in 2015. Since 2016, he has been a Professor with the Center of Technological Development (CDTec), UFPel, and a Researcher with the Group of Architectures and Integrated Circuits and the Video Technology Research Group. His main research interests include visual signal processing, low-complexity image and video coding, and digital design for multimedia systems. He is a member of the IEEE SPS and CAS societies and the Brazilian Computer Society.

DANIEL PALOMINO (Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the Federal University of Rio Grande do Sul (UFRGS) in 2013 and 2017, respectively. He is currently a Professor with the Center for Engineering, Federal University of Pelotas (UFPel), Brazil, and a Research Member of the Group of Architectures and Integrated Circuits and the Video Technology Research Group. He is also with the Postgraduate Program in Computer Science, UFPel, where he serves as an Advisor for Master's and Ph.D. students. His research experience includes six months as an Intern Researcher with the Karlsruhe Institute of Technology, Germany, and 11 months as a Visiting Professor with the University of Lisbon. His main research interests include power efficient computing systems, hardware architectures, and algorithms for image and video coding.

MARCELO PORTO (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2008 and 2012, respectively. He is currently a Professor with the Federal University of Pelotas (UFPel), Brazil, and a member of the Video Technology Research Group and the Group of Architectures and Integrated Circuits. He is currently the Coordinator of the Postgraduate Program in Computing, UFPel. He also has been holding the status of CNPq (National Council for Scientific and Technological Development) Productivity Research Fellow, since 2016. His research interests include video coding, motion estimation algorithms, point cloud compression, coding complexity reduction, and energy-efficient VLSI design for video coding.

BRUNO ZATT (Senior Member, IEEE) received the B.S. and M.S. degrees in computer engineering and the Ph.D. degree (*summa cum laude*) in microelectronics from the Federal University of Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 2006, 2008, and 2012, respectively. He currently is a Professor with the Federal University of Pelotas (UFPel), Pelotas, Brazil, and a member of the Group of Architectures and Integrated Circuits and the Video Technology Research Group. He has more than 16 years research experience on algorithms and hardware architectures for video processing, including three years as an Intern Researcher with the Karlsruhe Institute of Technology, Karlsruhe, Germany, and experience as a Visiting Professor with the University of California at Irvine, Irvine, USA. He has published over 100 papers in international journals/conferences and one book named *3D video coding for embedded devices*. He is a member of the IEEE CASS Visual Signal Processing and Communications Technical Committee, and Associate Editor for the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY. Since 2017, he has been holding the status of CNPq Productivity Research Fellow.

LUCIANO AGOSTINI (Senior Member, IEEE) received the M.S. and Ph.D. degrees in computer science from the Federal University of Rio Grande do Sul (UFRGS), Brazil, in 2002 and 2007, respectively. Since 2002, he has been a Professor with the Federal University of Pelotas (UFPel), Brazil, where he leads the Video Technology Research Group and the Group of Architectures and Integrated Circuits. He is a Advisor with the UFPel Master and Doctorate in Computer Science courses. He was the Executive Vice President for Research and Graduate Studies with UFPel from 2013 to 2017. He is a Brazilian Distinguished Researcher through a CNPq PQ-1D Grant. He has more than 300 published papers in respected international journals and conferences. His research interests include 2-D and 3-D video coding, algorithmic optimization, arithmetic circuits, and dedicated hardware design. He is an Associate Editor for the IEEE TRANSACTION ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the IEEE OPEN JOURNAL OF CIRCUITS AND SYSTEMS. He is a Senior Member of ACM, and a member of SBC and SBMicro Brazilian societies. He is also a member of the IEEE SPS, CS, and CAS societies and at CAS he is a member of the MSA and VSPC Technical Committees.