

In-loop Frame Super-resolution in AV1

Urvang Joshi, Debargha Mukherjee, Yue Chen, Sarah Parker, Adrian Grange
Google, USA.

Emails: {urvang, debargha, yuec, sarahparker, agrange}@google.com

Abstract— AV1 is a recently standardized royalty-free video codec from the industry consortium Alliance for Open Media. One of the most innovative coding tools supported in AV1 is an in-loop frame super-resolution mode, that allows an encoder to code any frame at a horizontally reduced spatial resolution by one of several levels, followed by upsampling and super-resolving to full resolution, before replacing reference buffers. This mode is partly enabled by a feature in AV1 that natively allows the motion compensated prediction loop to operate across scales between a coded frame and the available references, thereby allowing on-the-fly resolution change mid-stream within a sequence. For the actual super-resolving process a normative upscaler is followed by an in-loop restoration tool that recovers some of the high frequency information lost in the downsampling process. On-the-fly resolution change capability in conjunction with the frame-superresolution mode in AV1 opens up a new dimension for codec bitrate and quality optimization that has not been possible to explore in any prior standardized video codec with (soon expected) decoding hardware support. This paper provides an overview of how the relevant tools work in AV1, but unlocking them with intelligent encoder decisions to extract real-world benefit, is largely left as future work.

Index Terms—Video Compression, AV1, Alliance for Open Media, Super-resolution, Restoration, Downsampling

I. INTRODUCTION

The last decade has seen an explosive growth in the delivery and consumption of online video content, so much so that about 60-70% of all Internet bandwidth today is consumed by video. The ever increasing demand for higher resolution and higher quality video for both VOD and conversational video applications, as well as for newer applications such as gaming and VR, puts enormous demands on available network bandwidth. To cope with these demands, the pace of advancement in codec technology has also accelerated not just in traditional MPEG/ITU-T circles, but also in royalty-free codec projects and consortiums. MPEG standardized HEVC [1] in 2013 and is expected to finish work on the next generation codec VVC (Versatile Video Coding) [2] in 2020. Both of these codecs are royalty-bearing. On the other hand, the royalty-free WebM project [3] released VP9 [4] in 2013, and a recently formed industry consortium entitled Alliance for Open Media [5] released its first royalty-free codec AV1 [6] in 2018. AV1 is the latest and most advanced standardized video codec available today that is also royalty-free. This paper describes a specific feature in the codec that allows frames to be coded at lower resolution and then super-resolved in the coding loop.

In recent years, there has been a plethora of work on exploring adaptive downsampling based video coding. In such methods, frames or shots of a video that have low spatial detail are coded at lower resolution and then super-resolved on the decoder side - thereby achieving both superior rate-distortion efficiency and perceptual quality especially at very low bitrates, as well as lower encode and

decode complexity. The critical decisions in these methods are which frames or shots to downsample and by how much, and the method used for super-resolving on the decoder side. Note that in this paper *shot* refers to a set of frames starting from a key-frame or IDR frame that can be decoded as an independent sequence.

One class of these methods perform the downsampling and upsampling operations non-normatively as pre- and post-processing steps outside the codec so that the actual bit-stream for a shot remains compliant to a standard. These methods are also naturally compatible with today's pre-eminent streaming architectures such as segmented adaptive bitrate switching (DASH, HLS, etc.), where each segment or shot can be switched to a different resolution and/or bitrate depending on current network characteristics and the resolution desired. The viewing device is then expected to rescale every segment it decodes to the display viewport resolution before displaying. Note that no normative change in the codec or the transport mechanism is required in this case, since each segment is a self-contained fully compliant bit-stream. This strategy is followed by pretty much all video streaming services today including YouTube and Netflix.

Several of these works [7]-[12] deal with optimal decision making for the best downsampling ratio for an entire shot delimited by keyframes. Among non codec compliant variants of the shot based downsampling approach, some design downsampling and normative upsampling filters [7] specific to the downsampling ratios used. Some adaptive versions send additional side-information in the bitstream to signal how much to upsample spatially [9] and in some cases temporally [12], for each shot, or the specific upsampling/super-resolving method to use [11] for each frame and each block within it based on a classification map conveyed in the bitstream. Further, instead of interpolating with a linear filter (Bicubic, Lanczos, etc.), more sophisticated super-resolving techniques such as CNN based single-frame ones [10][12] have been used. However, in all these methods, the main motion compensated prediction loop requires a coded frame and reference frames to operate at the same resolution.

A second class of downsampling based video coding methods allow more flexibility within a coded shot (sequence) where certain Inter frames can be coded at a lower resolution even though the references may be at higher resolution. The low-resolution frames are then upsampled to full-resolution at a suitable stage in the coding pipeline depending on the specific architecture. The difficulty with supporting such a mode of operation is that the core motion compensated prediction loop must be modified/enhanced to support predicting a frame from references that may be at different resolutions. Some of the earliest work in this class of methods [13]-[16] used a mixed resolution architecture where periodic high-resolution frames are interspersed with a set of low resolution frames. These methods were explored in the context of side-information generation for Wyner-Ziv coding [13] and key-frame based superresolution [14]-[16], where in all cases multi-frame motion based methods were used for super-resolving. In some of the mixed resolution approaches, the periodic high-resolution (HR) frames are all key frames, and in order to code

the low-resolution (LR) frames in a period, the reconstructed key-frame is downsampled normatively to the same resolution as the LR frames and stored in the reference frame buffer before applying motion compensation. A more general approach that would remove the requirement for HR frames to be key frames would be to have separate coding loops and reference frame stores for different resolutions. For example if there are two resolutions supported in an architecture, there could be two loops: (i) one for HR frames that are inter coded using other HR reconstructions, along with LR reconstructions that are normatively upsampled and/or super-resolved; and (ii) another for LR frames that are inter coded using other LR reconstructions along with HR reconstructions that are normatively downsampled. To our knowledge there has been little prior work exploring these architectures for video coding.

In AV1, we adopt a fundamentally different approach, where we natively support predicting across scales in the motion compensation loop. Note that VP9 also supported across-scale prediction, but the mechanism has been significantly improved in AV1. With this feature, at any point in the encoding process, the encoder may choose to step down or up on-the-fly to lower or higher resolution without requiring a keyframe. Consequently, each reference buffer may be at a different resolution from a coded frame, and the motion compensation process handles the scale difference naturally. This feature by itself allows AV1 to support on-the-fly resolution change that most other codecs do not support. In addition, this feature provides a mechanism for supporting a frame-level in-loop super-resolution mode in the bit-stream syntax. Specifically, if the super-resolution mode is enabled for a frame, the frame is non-normatively downsampled to a lower resolution and coded at lower resolution using higher resolution references in the motion compensation loop, followed by normative upsampling and super-resolving to generate a full resolution output frame that is displayed and that updates a reference frame store slot for use in subsequent frames. Note also that high-performant CNN based and/or multi-frame restoration methods are quite complex for today's video decoding hardware architectures. Therefore, AV1 uses a normative linear upscaler followed by reusing an existing in-loop filtering tool called Loop Restoration for the purpose of super-resolving. Furthermore, in order to keep line-buffer requirements in the hardware decoding pipeline small, only subsampling and super-resolving in the horizontal dimension is supported in AV1's in-loop super-resolution mode, but on-the-fly resolution change can still use scaling in both dimensions. The goal of this paper is to provide a holistic description of the super-resolution tool in AV1 and explain how the tool works. Some preliminary results are provided, but extracting real value out of the tool by good decisions on the encoder side for measurable rate-quality trade-offs is still a work in progress.

The rest of the paper is organized as follows. Section II provides a detailed description of the relevant tools, including scaled prediction, the actual frame super-resolution mode, the normative horizontal only upscaler and the loop-restoration tool. Section III provides some results showing the benefit of the super-resolving process as opposed to just linear upscaling, and some preliminary results for use of the tool in actual encoding. Section IV concludes the paper.

II. DETAILS

In sub-section A below, we provide a brief overview of the scaled prediction framework in AV1. This forms the basis for the super-resolution tool, described in sub-section B.

A. Scaled Prediction

In order to enable the codec to switch frame resolutions mid-stream, both AV1 and its predecessor VP9 support the ability to predict across scales in the inter prediction loop. As shown schematically in Fig. 1, this allows any frame or frames to be non-normatively downsampled or upsampled (Fig. 1 shows downscaling only) on-the-fly before encoding at a different resolution. The reconstructed frame after encoding at the reduced or increased resolution then replaces one of the reference buffer slots at that resolution. Therefore, at any point during the encoding and decoding process, any inter frame could be predicted from references that are at different resolutions, and consequently a normative mechanism to predict a block in that frame from a different resolution reference buffer needs to be defined. In principle, as long as we have defined a normative upscaler and a normative downscaler, such prediction across scales would be possible to support. However it would be more compute efficient to combine such rescaling with subpel interpolation for motion compensation, and that is what AV1 does.

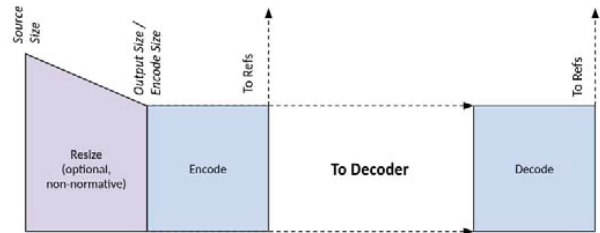


Fig. 1. Scaled prediction framework

Fig. 2 shows a scenario where a 4x4 block from the source needs to be predicted from a different resolution reference buffer. The motion vectors transmitted in the bitstream are always at the source resolution. So they are first scaled up or down based on the resolution ratio between the reference and source, and the corresponding source block pixels projected on the reference grid can then be obtained, as shown on the right of Fig. 2. Note that the relative sub-pixel positions horizontally (vertically) on the reference grid are the same in each row (column). Hence, the interpolation for scaled prediction can be implemented simply as separable filtering in each dimension using a suitable starting sub-pixel offset and a sub-pixel step between pixels.

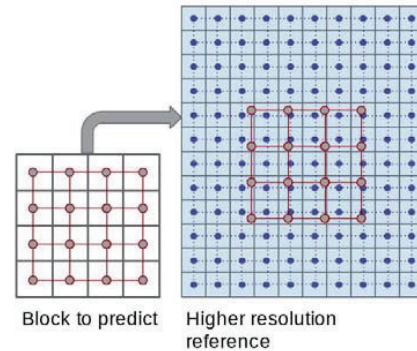


Fig. 2. Predicting a block from different (higher shown) resolution reference

For sub-pixel interpolation, AV1 uses the same $\frac{1}{16}$ th phase 8-tap filters that are used for normal unscaled prediction. However, some adjustments are needed to improve the coding efficiency. Due to the fact that the start offset and step size cannot have infinite precision,

the error between real subpel position and integerized subpel position can grow large near the end of the block as shown in Fig. 3, thereby causing poor prediction and loss in coding efficiency. To reduce this error, AV1 uses a much higher precision ($\frac{1}{1024}$ th pel) for start offset and step size even though the actual filters are still $\frac{1}{16}$ th phase. For each source pixel, the high resolution offset corresponding to it on the reference grid is rounded down to the nearest $\frac{1}{16}$ th pel position to obtain the specific filter phase to be used. Incidentally, VP9 used $\frac{1}{16}$ th precision for start offset and step that was clearly inadequate and inefficient. For practical reasons, AV1 imposes a limit on the scale ratio allowed between a source frame and any of the references. Specifically, a reference cannot be more than 2 times or less than $\frac{1}{16}$ times the source resolution in any dimension. As long as this constraint is satisfied, any frame in a sequence can be coded at an arbitrary resolution.

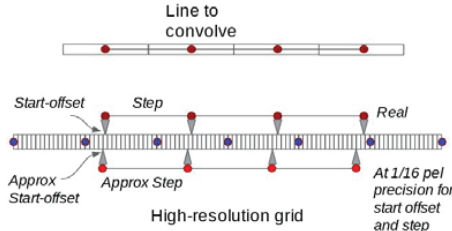


Fig. 3. Accumulation of error near end of blocks if lower (1/16-pel) precision is used for start offset and step. With higher (1/1024-pel) precision the error accumulation is much smaller.

B. Frame Super-resolution Mode

Now that we know that the motion compensation loop in AV1 can predict across scales, we can proceed to describing the super-resolution coding mode in the context of the overall in-loop filtering pipeline in AV1.

B.1 Framework At a Glance

Fig. 4 depicts the overall in-loop filtering pipeline in AV1 and the super-resolution framework. According to this framework, on the encoder side, a source frame may be first downsampled non-normatively and encoded at a lower resolution. The deblocking filter, and another filter called Constrained Directional Enhancement Filter (CDEF) [17] supported in AV1, are then used to remove blocking and ringing artifacts while preserving edges, at the lower resolution. This is followed by a normative Linear Upsampling

process to full source resolution. Finally, another AV1 tool called Loop Restoration (LR) [18] is applied at full resolution to recover the high frequencies lost during the prior downscaling and quantization processes. The loop restored frame updates the reference buffer store for encoding subsequent frames. A similar process is followed on the decoder side as well. Note that the Deblocking, CDEF, Linear Upscaling and LR in order are all part of the normative in-loop filtering process in AV1, where each filter can be turned on or off per frame.

The super-resolution mode is a special mode signaled at the frame level to indicate whether the Linear Upscaler in particular, is to be used, and what upscaling ratio if used is to be applied. The other filters stay optional, but LR, if used in addition, will enable smart super-resolving instead of just upscaling. Even though the original super-resolution mode proposal in AV1 allowed downscaling and upscaling a frame in both dimensions, in the final AV1 standard, the upscaler is restricted to operate only horizontally in order to enable cost-effective hardware decoding without extra line buffer usage. In Fig. 4, we have annotated the frame resolution at different stages for an example source of resolution 4K x 2K, with down-scaling ratio of $\frac{1}{2}$ horizontally.

Specifically, each coded frame in the AV1 bit-stream signals its output frame dimensions $W \times H$ along with a horizontal upscaling ratio to be used by the Linear Upscaler in the in-loop filtering pipeline. The upscaling ratio can have 9 possible values: $d/8$, with $d = 8, 9, \dots, 16$, where $d = 8$ corresponds to no upscaling. The corresponding downsampling ratios before encoding would then be $8/d$. Given the output frame dimensions $W \times H$ and the upscaling ratio d , both encoder and decoder can calculate the low-resolution coded frame dimension as: $w \times H$, where reduced width $w = \frac{8W + d/2}{d}$, where $d/2$ is added to round to the nearest integer. After reconstructing the coded low-resolution frame, deblocking and CDEF are also applied at low-resolution. The Linear Upscaling process that follows, takes in the frame at resolution $w \times H$ and guarantees outputting a frame of dimensions exactly the same as the output frame dimensions $W \times H$ specified in the frame header. LR is next applied at the $W \times H$ output resolution.

Note that the AV1 bit-stream is flexible enough that each coded frame can use (a) horizontal-only super-resolution mode as described above, (b) code a frame at an arbitrary resolution within the ratio constraints, (c) a combination of both, or (d) no scaling at all. An intelligent encoder can utilize these options as additional knobs to be

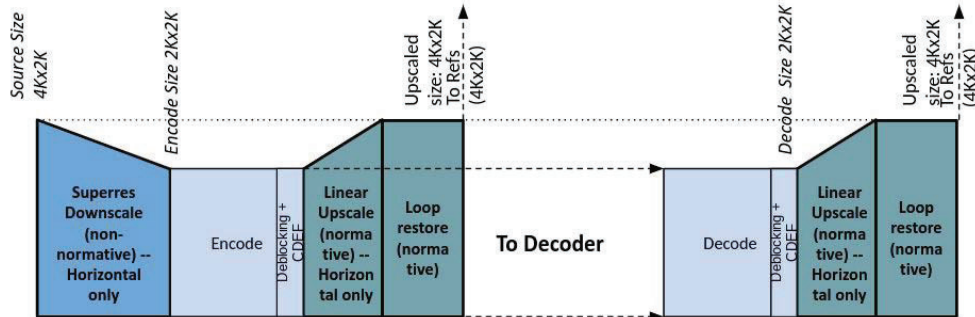


Fig. 4. Frame Super-resolution Framework

tuned, over the usual quantization knob, for targeting desired quality-bitrate tradeoffs.

B.2 Normative Linear Upscaling

The normative horizontal linear upscaler in AV1 uses a $\frac{1}{16}$ th phase linear 8-tap filter for interpolation of each row. This is similar to the scaled prediction described in Section II A except that it is applied only horizontally. Given the coded frame width w and the output frame width W , the step size for interpolation is computed as the ratio w/W at high enough precision. Note that even though this value is close to $8/d$, it is not exactly the same because the output width is not necessarily a multiple of d . The step size and start offsets for upscaling interpolation use an even higher $\frac{1}{16384}$ th pixel precision than what is used for scaled prediction, since the width of a frame can be much larger than what is needed for scaled block prediction leading to larger error accumulation.

When scaling a row of pixels in a frame to a higher resolution, a naive approach is to keep the start offset (center of the first output pixel) aligned with the center of the first source (input) pixel. However, AV1 applies a couple of smart adjustments to the start offset as described below.

Fig. 5 motivates the first adjustment. The AV1 upscaler is designed to ensure that the overall length covered by the input pixels is the same as that covered by the output pixels. Therefore if the w input pixels are each of unit length, the W output pixels covering the same overall length are each of length $\frac{w}{W}$. Assuming the leftmost pixel of the low resolution input on the low-resolution grid is at location 0, the leftmost pixel location of the high resolution sequence on the low-resolution grid is at: $-(\frac{W-w}{2W})$. This value, which is negative since $W > w$, is the primary determinant of the starting offset adjustment for the interpolation.

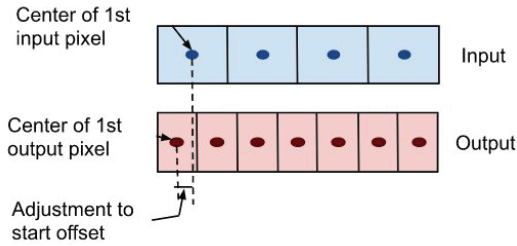


Fig. 5 Start offset determination

The second adjustment is motivated by the fact that, even at such a high (but finite) precision for starting offset and step size, there will be a modest error accumulation near the end of the sequence. Maximum error accumulation will be after the last pixel, and is given by: $e = W \times \text{FinitePrecision}(w/W) - w$. Centralizing this error by shifting the start offset by another $\frac{e}{2}$ is beneficial for interpolation. So, the final adjustment to the start offset is given by:

$$-(\frac{W-w}{2W}) - \frac{e}{2}$$

B.3 Loop Restoration

The Loop Restoration tool [18] in AV1 introduces a switchable framework for in-loop restoration, where for each restoration unit of size 256×256 or 128×128 or 64×64 that a frame is divided into, one of three supported restoration filters may be used. The bitstream signals the map for filter usage along with associated

parameters for each. Available restoration filters are (a) Separable Symmetric Wiener Filter (b) Dual Self-guided Filter with Subspace Projection and (c) No Filter.

Specifically, the Wiener filter is particularly good at super-resolving and recovering lost high frequencies, where the encoder uses a least-squares method to determine the optimal filter coefficients from the upscaled frame and the source frame, and conveys them in the bitstream. The dual guided filter can also recover lost frequencies when an appropriate search and least-squares based subspace projection is conducted on the encoder side. Therefore, when the superresolution mode is signaled in the bitstream, it is highly recommended that the loop-restoration tool is also used, even though that is not enforced in the bit-stream syntax. In fact, without this tool, there is no super-resolving, only upscaling in AV1.

III. EXPERIMENTAL RESULTS

We demonstrate the benefit of superres by testing its coding performance for various scenarios in rate vs. PSNR obtained with AV1 (Aug 2019 version). The tests are performed on three test sets as follows, each containing standard videos in 4:2:0 8bit format:

- hdsres set: 38 videos at 720p and 1080p resolutions
- midres set: 30 videos at 480p and 360p resolutions
- lowres set: 38 videos at CIF and SIF resolutions

The libaom encoder (the open-source reference encoder for AV1) is set to use constant quality mode with ‘--end-usage=q’ option, with the corresponding --cq-level={8, 12, 17, 23, 29, 36, 44, 53, 63} covering the whole range of qualities.

The coding performance comparisons are represented by BDRate. A negative BDRate means using less bits to achieve the same quality. PSNR is the objective metric used to compute the BDRate. The exact encoding options used are noted for each scenario below.

A. Super-resolution Without and With Loop Restoration

In this experiment, we quantify the benefit of the ‘Loop Restoration’ operation itself (the last operation in Fig. 4), when using super-resolution. Baseline runs use these options in the libaom encoder to enable super-resolution with a fixed scale for all frames, but disable loop restoration: ‘--cpu-used=1 --superres-mode=1 --superres-denominator=d --superres-kf-denominator=d --enable-restoration=0’ (with d values 9, 10, 12, 14, 16). These are compared against test runs with same parameters except --enable-restoration=1. Average BDRate improvement over the entire test sets for various d values are shown in Table 1.

Table 1. BDRate Improvement (%) With Loop Restoration

Scaling Denominator (d)	lowres	midres	hdsres
16	-14.43	-13.03	-11.77
14	-9.99	-7.26	-8.38
12	-9.82	-6.95	-6.87
10	-13.8	-8.68	-6.52
9	-8.66	-5.78	-4.18

We can observe that there is a significant compression improvement when using Loop Restoration, and the benefit increases as the scaling factor increases.

B. Full-resolution vs Super-resolution: Image Coding

In this test, we demonstrate the benefit of using super-resolution for image coding scenario. For this, we encode 20 frames of h265 set, all encoded as keyframe. The baseline run uses these parameters in the libaom encoder to encode at full-resolution: ‘--cpu-used=0 --kf-min-dist=0 --kf-max-dist=0’. The test run additionally uses ‘--superres-mode=4’, which adaptively chooses which frames should use super-resolution and the appropriate scaling factor, based on horizontal frequency analysis. Overall coding improvement and clips with largest improvements are shown in Table 2.

C. Full-resolution vs Super-resolution: Video Coding

Similar to B, but now we test super-resolution for video coding scenario. For this, we encode 20 frames of h265 set, with 1 keyframe and other inter frames. The baseline run uses these parameters with the libaom encoder to encode at full-resolution: ‘--cpu-used=0’. The test run additionally uses ‘--superres-mode=4’ to intelligently pick super-resolution as in B. Clips with largest improvements are shown in Table 3. Note that we focus on clips with most improvements, as we want to showcase the potential of super-resolution framework, and not the particular heuristics currently used by the libaom encoder.

Table 2. BDRate Improvement For Image Coding

Test Set / Clip	BDRate (%)
Overall	-0.206
sunflower_720p25.y4m	-3.732
cyclists_720p30.y4m	-1.782
jets_720p30.y4m	-1.441
tractor_1080p25.y4m	-1.387
riverbed_1080p25.y4m	-1.344
basketballdrive_1080p50.y4m	-1.107
ped_1080p25.y4m	-0.858
rush_hour_1080p25.y4m	-0.517

Table 3. BDRate Improvement For Video Coding

Test Set / Clip	BDRate (%)
sunflower_720p25.y4m	-0.892
riverbed_1080p25.y4m	-0.770
cyclists_720p30.y4m	-0.502
crew_720p30.y4m	-0.248
kimono1_1080p24.y4m	-0.247

IV. CONCLUSION

In this paper, we provided an overview of the in-loop super-resolution coding mode in AV1, including some key components such as across-scale prediction, horizontal-only upscaling and loop restoration, that make it both efficient and practical for real-world use. We provide some results showcasing the benefit of superresolution with loop restoration vs. only upscaling. Further we provide some initial results using a simple method to adaptively choose super-resolution mode on the encoder side.

However note that there are many more possibilities to explore on the encoder side to intelligently choose this mode for a variety of scenarios, and that will comprise our future work in this area.

V. REFERENCES

- [1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) standard", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649-1668, Dec. 2012.
- [2] <https://mpeg.chiariglione.org/standards/mpeg-i/versatile-video-coding>, Versatile Video Coding.
- [3] <https://www.webmproject.org/>, WebM: an open web media project.
- [4] D. Mukherjee, J. Bankoski, A. Grange, J. Han, J. Koleszar, P. Wilkins, Y. Xu, R.S. Bultje, "The latest open-source video codec VP9 - an overview and preliminary results", *Proc. Picture Coding Symposium (PCS)*, December 2013.
- [5] <https://aomedia.org/>, Alliance for Open Media.
- [6] Y. Chen, D. Mukherjee, et. al, "An Overview of Core Coding Tools in the AV1 Video Codec," *Proc. Picture Coding Symposium*, San Francisco, CA, USA, June 2018.
- [7] J. Dong, Y. Ye, "Adaptive downsampling for high-definition video coding", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 3, pp. 480-488, Mar. 2014.
- [8] R.-J. Wang, C.-W. Huang, P.-C. Chang, "Adaptive downsampling video coding with spatially scalable rate-distortion modeling", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 11, pp. 1957-1968, Nov. 2014.
- [9] D. Barreto, L. D. Alvarez, R. Molina, A. K. Katsaggelos, G. M. Callic, "Region-based super-resolution for compression", *Multidimensional Syst. Signal Process.*, vol. 18, pp. 59-81, 2007-Sep.
- [10] Mariana Afonso, Fan Zhang, Angeliki Katsenou, Dimitris Agrafiotis and David Bull, "Low complexity video coding based on spatial resolution adaptation," *Proc. IEEE Int. Conference on Image Processing (ICIP)*, Aug 2017.
- [11] G. Georgis, G. Lentaris, D. Reisis, "Reduced complexity superresolution for low-bitrate video compression", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 2, pp. 332-345, Feb. 2016.
- [12] Mariana Afonso, Fan Zhang, David R. Bull, "Video Compression Based on Spatio-Temporal Resolution Adaptation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 29, no. 1, pp 275-280, Jan 2019.
- [13] D. Mukherjee, B. Macchiavello, R. L. de Queiroz, "A simple reversed complexity Wyner-Ziv video coding mode based on a spatial reduction framework", *Proc. SPIE VCIP*, vol. 6508, pp. 1-12, Jan. 2007.
- [14] Edson Mintsu Hung, Ricardo L. de Queiroz, Fernanda Brandi, Karen Franca de Oliveira, Debargha Mukherjee, "Video Super-Resolution Using Codebooks Derived From Key-Frames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1321-31, Sept. 2012.
- [15] Brandi, R. L. de Queiroz, D. Mukherjee, "Super-resolution of video using key-frames and motion estimation", *Proc. IEEE Int. Conference on Image Processing (ICIP)*, pp. 321-324, 2008-Oct.
- [16] M. Shen, P. Xue, C. Wang, "Down-sampling based video coding using super-resolution technique", *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 6, pp. 755-765, Jun. 2011.
- [17] S. Midtskogen, J.-M. Valin, The AV1 Constrained Directional Enhancement Filter (CDEF), *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.
- [18] D. Mukherjee, S. Li, Y. Chen, A. Anis, S. Parker, J. Bankoski, "A Switchable Loop-restoration with Side-information Framework for The Emerging AV1 Video Codec," *Proc. IEEE Int. Conference on Image Processing (ICIP)*, Beijing, China, Sep. 2017.