ADAPTIVE RESOLUTION CHANGE USING UNCODED AREAS AND DICTIONARY LEARNING-BASED SUPER-RESOLUTION IN VERSATILE VIDEO CODING

Jens Schneider* *Johannes Sauer** *Christian Rohlfing**

*Institut für Nachrichtentechnik
RWTH Aachen University

ABSTRACT

The concept of Adaptive Resolution Change (ARC) in video coding is already known from former international standards such as MPEG-4 [1]. However, in MPEG-4 linear filters are used for upsampling, which is crucial to coding video at varying resolution. With the rise of machine learning-based super-resolution methods in the last decade, powerful algorithms outperforming conventional upsampling methods were developed. This contribution introduces an ARC concept using uncoded areas within a frame of a video sequence and a Dictionary Learning (DL)-based Super-Resolution (SR) scheme. In this concept, a frame contains a picture at different resolution levels, which are spatially separated by the use of slices and tiles. Generally, a tile can be marked as coded or uncoded. Slices which do not hold any coded tiles are omitted from the bitstream. Thus, only one resolution level needs to be coded, while the other is generated at the decoder side. At the encoder a rate-distortion decision is made in order to decide, which resolution level should be coded. Simulation results show that gains with respect to the Versatile Video Coding (VVC) standard in development can be achieved at low bitrates.

Index Terms— video compression, adaptive resolution change, dictionary learning, versatile video coding

1. INTRODUCTION

The concept of changes in resolution when encoding image or video content has been studied in different standardization [1] and recent research activities [3], [2], [4]. Moreover, in adaptive video streaming the change of resolution is a key concept for the estimation of the convex hull of the rate-distortion curve [5]. It was observed that at low bitrates the processing chain of downsampling – coding – upsampling outperforms coding the video at full resolution in terms of rate-distortion measures. However, recent and frequently used video coding standards such as AVC [6] and HEVC [7] do not entail coding tools allowing a change of resolution within the coding loop. Recent research showed that there is still potential for better coding performance when Coding Tree Units (CTUs) are downsampled before coding in an All Intra (AI) coding scenario [2]. Moreover, ARC is considered as a useful tool for video telephony and is investigated in the current standardization activity towards VVC [8].

Generally, the concept of ARC can be seen as a version of steered quantization since downsampling effectively discards frequencies above half of the new sampling rate when anti-aliasing is performed with an appropriate filter. Therefore, a lower rate can be expected when coding is performed at a lower resolution. Besides of frequency-zeroing, a lower resolution results in less side information which needs to be transmitted. This has the effect that less partitioning information needs to be sent, when the resolution is lowered. Due to the lower rate needed for side information and high frequency content, the encoder can spend more bits for the downsampled video content itself. This leads to the result that the lower resolution video can be coded at a lower value of the Quantization Parameter (QP).

This contribution introduces an ARC-concept using uncoded areas within a frame and DL-based SR. The rest of the paper is organized as follows. In Section 2, the fundamentals of downsampling and upsampling are briefly reviewed. Section 3 deals with the concept of ARC using uncoded areas. In Section 4, the simulation setup and results are presented and Section 5 concludes the paper.

2. DOWNSAMPLING AND UPSAMPLING

2.1. Resampling with Linear Filters

The process of downsampling is a linear but not space invariant operation which resamples a discrete signal $\boldsymbol{s}$ at a lower rate. Since downsampling generally introduces alias for critically sampled signals, the signal is typically processed by an anti-aliasing filter $\boldsymbol{h}_\downarrow$. The filtering and downsampling operation can be expressed by

$$\boldsymbol{s}_\downarrow[n] = \boldsymbol{R}_\downarrow \left(\boldsymbol{h}_\downarrow * \boldsymbol{s}[n]\right) \tag{1}$$

with the downsampling matrix $\boldsymbol{R}_\downarrow$ and the convolution operator $*$. Upsampling is defined by introducing zeros between the samples of a discrete signal. This operation causes periodic copies of the signal's baseband in the frequency domain. In order to interpolate the positions where zeros were inserted, the signal is convolved with an interpolation filter $\boldsymbol{h}_{\uparrow}$. The upsampling equation reads with the upsampling matrix $\boldsymbol{R}_{\uparrow}$

$$\boldsymbol{s}_{\uparrow}[n] = \boldsymbol{h}_{\uparrow} * (\boldsymbol{R}_{\uparrow}\, \boldsymbol{s}[n])\,. \tag{2}$$

Note that for multidimensional signals the downsampling and upsampling operation can be applied in any direction independently.

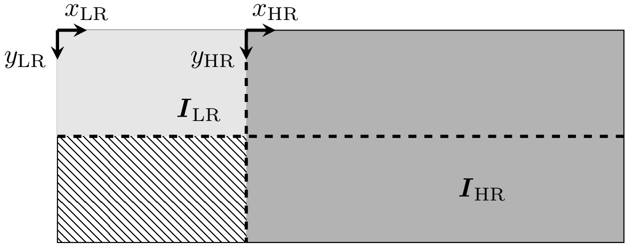

Fig. 1. Picture Level ARC-scheme. The hatched area in uncoded. The light gray area contains the half resolution video sequence. The dark gray area holds the video in full resolution. Only one resolution level is coded. The other area is marked as uncoded and inferred at the decoder side. Dashed lines indicate slice/tile boundaries.

2.2. Dictionary Learning-based Super-Resolution

Dictionary learning-based super-resolution builds on the assumption that image signals can be represented sparsely in a set of K learned functions called a dictionary, i.e.

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}. \tag{3}$$

Thereby $\boldsymbol{x} \in \mathbb{R}^{s_{\mathrm{p}}^2}$ represents a vectorized image patch containing s_{p}^2 pixels, $\boldsymbol{D} \in \mathbb{R}^{s_{\mathrm{p}}^2 \times K}$ represents the dictionary and $\boldsymbol{\alpha} \in \mathbb{R}^{K}$ is the sparse coefficient vector holding an approximate representation of the image patch with the model error $\boldsymbol{\varepsilon} \in \mathbb{R}^{s_{\mathrm{p}}^2}$. For most image processing applications the patches overlap so that blocking artefacts are reduced. In the patch combination process, pixels covered by more than one patch are reconstructed by averaging over all estimates. A typical step sizes for patch extraction is $s_{\mathrm{s}} = 2$ in case of SR with a scaling factor of $s_{\downarrow\uparrow} = 2$, i.e. patches are extracted at pixel positions of the Low Resolution (LR) image. If the dictionary is known, the coefficients $\boldsymbol{\alpha}$ can be obtained solving, e.g.

$$\boldsymbol{\alpha} \leftarrow \arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 \tag{4}$$

with the penalization parameter λ on the l_1-norm of the solution steering the sparsity in $\boldsymbol{\alpha}$. The optimization problem (4) can be solved by e.g. the LARS-algorithm [9].

In order to apply DL / Sparse Coding (SC) to inverse problems like SR, an approach using two dictionaries with coupled sparsity has become state of the art [10], [11], [12]. This approach will be introduced in the following. Preprocessing steps are neglected for simplicity in the following. Let

$$\boldsymbol{x}_{\mathrm{LR}} = f\left(\boldsymbol{x}_{\mathrm{HR}}\right)$$

be the LR image patch degenerated by the function $f(\cdot)$. In case of the SR problem, the function $f\left(\cdot\right)$ maps a High Resolution (HR) image patch to its upsampled LR representation. Therefore, $\boldsymbol{x}_{\mathrm{HR}}$ and $\boldsymbol{x}_{\mathrm{LR}}$ are of the same dimensions and differ only in the frequency distribution of their content. Put differently, patch pairs $\{\boldsymbol{x}_{\mathrm{HR}}, \boldsymbol{x}_{\mathrm{LR}}\}$ can be generated by downsampling and upsampling an image and extracting patches from the original and the degenerated one. Sourcing those patch pairs from a training set of natural images, the dictionaries can be trained solving

$$\boldsymbol{D}_{\mathrm{LR}} \leftarrow \arg\min_{\boldsymbol{D}_{\mathrm{LR}}} \sum_{i=1}^{n} \frac{1}{2}\|\boldsymbol{x}_{\mathrm{LR},i} - \boldsymbol{D}_{\mathrm{LR}}\boldsymbol{\alpha}_i\|_2^2 + \lambda\|\boldsymbol{\alpha}_i\|_1, \tag{5}$$

$$\boldsymbol{D}_{\mathrm{HR}} \leftarrow \arg\min_{\boldsymbol{D}_{\mathrm{HR}}} \|\boldsymbol{X}_{\mathrm{HR}} - \boldsymbol{D}_{\mathrm{HR}}\boldsymbol{A}\|_2^2, \tag{6}$$

with $\boldsymbol{X}_{\mathrm{HR}} = [\boldsymbol{x}_{\mathrm{HR},1}, \boldsymbol{x}_{\mathrm{HR},2}, ..., \boldsymbol{x}_{\mathrm{HR},n}]$, $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_n]$ and n being the number of training patches. Thereby $\boldsymbol{D}_{\mathrm{LR}}$ represents a dictionary which was trained for sparse representations of the degenerated signal $\boldsymbol{x}_{\mathrm{LR}}$. $\boldsymbol{D}_{\mathrm{HR}}$ is the dictionary with coupled sparsity used for the reconstruction. Note that the same coefficients vector $\boldsymbol{\alpha}$ is assumed to represent the signal $\boldsymbol{x}_{\mathrm{LR}}$ sparsely in $\boldsymbol{D}_{\mathrm{LR}}$ and the original signal $\boldsymbol{x}$ in $\boldsymbol{D}_{\mathrm{HR}}$ respectively. The inverse function $f^{-1}(\cdot)$ is then approximated by the coupled sparsity approach as follows:

$$\begin{aligned} &f^{-1}\left(\boldsymbol{x}_{\mathrm{LR}}\right) \approx \\ &\quad \boldsymbol{D}_{\mathrm{HR}}\left(\arg\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\boldsymbol{x}_{\mathrm{LR}} - \boldsymbol{D}_{\mathrm{LR}}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1\right). \end{aligned} \tag{7}$$

From (7) it can be derived that the inverse function consists of a non-linear analysis step and a linear reconstruction step. Both steps need to be performed for every extracted image patch. As already shown in [13] this function approximation can be interpreted as a convolutional neural network with a non-linear activation function other than the common sigmoid, Rectified Linear Unit (ReLU) etc. but a SC-based activation.

3. ADAPTIVE RESOLUTION CHANGE USING UNCODED AREAS

3.1. Overview

The concept of uncoded areas allows to have CTUs in a frame which are not coded but may be inferred at the decoder side. In this scheme, the scan order of CTUs is reorganized using tiles such that parts of the frame can be processed independently. Furthermore, tiles can be marked as coded or uncoded and slices which do not entail any coded tiles are not coded into the bitstream. This uncoded area may be filled by other means at the decoder.

Fig. 1 shows an example how uncoded areas can enable ARC. The frame containing the video at different resolutions

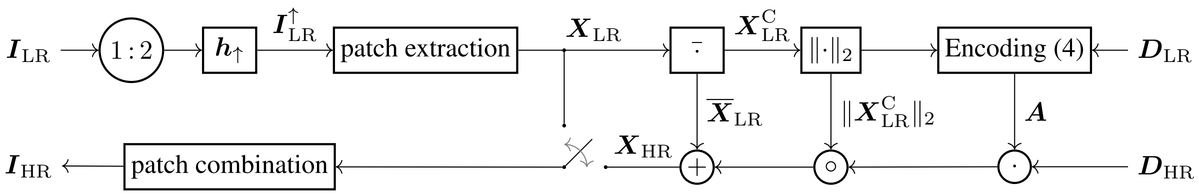
$\boldsymbol{I}_{\mathrm{LR}}$
1 : 2
$\boldsymbol{h}_{\uparrow}$
$\boldsymbol{I}_{\mathrm{LR}}^{\uparrow}$
patch extraction
$\boldsymbol{X}_{\mathrm{LR}}$
$\bar{\cdot}$
$\boldsymbol{X}_{\mathrm{LR}}^{\mathrm{C}}$
$\|\cdot\|_2$
Encoding (4)
$\boldsymbol{D}_{\mathrm{LR}}$
$\overline{\boldsymbol{X}}_{\mathrm{LR}}$
$\|\boldsymbol{X}_{\mathrm{LR}}^{\mathrm{C}}\|_2$
$\boldsymbol{A}$
$\boldsymbol{I}_{\mathrm{HR}}$
patch combination
$\boldsymbol{X}_{\mathrm{HR}}$
+
∘
·
$\boldsymbol{D}_{\mathrm{HR}}$

Fig. 2. DL-based SR scheme with the preprocessing steps of centering and normalization. The switch is controlled by the sparsity in $\boldsymbol{A}$. If there is e.g. no significant coefficient for a patch its upscaled LR representation is used for the reconstruction.

is organized such that the low resolution picture $\boldsymbol{I}_{\mathrm{LR}}$ is placed in the upper left corner of the frame and the high resolution picture $\boldsymbol{I}_{\mathrm{HR}}$ is placed next to it. The scaling factor between the two resolution levels was chosen to be $s_{\downarrow\uparrow} = 2$ in this work and the low resolution content was generated by using the zero-phase downsampling filter known from the scalable extension of HEVC [14], i.e.

$$\boldsymbol{h}_{\downarrow} = [2, -3, -9, 6, 39, 58, 39, 6, -9, -3, 2]/128.$$

The hatched area (▨) in the bottom left is always marked as uncoded as it does not hold any meaningful image content. Note that slice and tile boundaries are aligned and the low resolution image $\boldsymbol{I}_{\mathrm{LR}}$ is contained in one slice, whereas the high resolution image $\boldsymbol{I}_{\mathrm{HR}}$ is split by a slice boundary. This will generally lead to some overhead when the high resolution image is coded since prediction across slice boundaries is not allowed and the CABAC-engine is reset at slice and tile boundaries. However, this overhead arises due to the limitations of the VTM-5.0 reference software [15] since it does not allow to specify a single slice in a tile for the high resolution image. This would in general be allowed by the specification of slices and tiles in HEVC / the current state of VVC [16]. In order to signal which area of the frame was marked as coded, a flag in the slice header of the first slice of the picture is coded. Dependent on this flag either the low resolution image $\boldsymbol{I}_{\mathrm{LR}}$ or the high resolution image $\boldsymbol{I}_{\mathrm{HR}}$ needs to be inferred at the decoder side.

3.2. Inference of Uncoded Area and Rate-Distortion Decision

The encoder decides whether the full resolution picture or its lower resolution representation should be coded. The decision is dependent on both, the overall rate needed for the frame and the overall distortion measured in the high resolution image space. In order to perform the distortion measurement for the low resolution content a high resolution image needs to be generated. Generally, the coding architecture using uncoded areas allows for any inference method. We derive an interpolation filter from the HEVC half-pixel motion compensation filter, resulting in

$$\boldsymbol{h}_{\uparrow} = [-1, 0, 4, 0, -11, 0, 40, 64, 40, 0, -11, 0, 4, 0, -1]/64.$$

Moreover, we apply the DL-based refinement depicted in Fig. 2. Generally, this scheme is close to known DL-based SR algorithms like [11]. Centering $(\bar{\cdot})$, i.e. subtraction of the mean and normalization $(\|\cdot\|_2)$ to unit l_2-norm is performed for preprocessing of the image patches $\boldsymbol{X}_{\mathrm{LR}}$. In order to limit noise amplification for low variance patches, patches with norms $\|\boldsymbol{X}_{\mathrm{LR}}\|_2 < 0.05$ are not normalized. Within the Encoding block the SC problem (4) is solved and the resulting coefficients $\boldsymbol{A}$ are used for reconstruction. Note that the switch in the lower branch of the block diagram controls whether a patch is processed or the interpolated version of the LR patch $\boldsymbol{x}_{\mathrm{LR}}$ is directly bypassed into the set of reconstructed patches. The switch is controlled based on the sparsity of the solution $\boldsymbol{\alpha}$ for a patch and put to the upper position, if the support of the solution is empty or greater than the average support for a patch. The first case of an empty support would lead to a reconstruction of a flat patch and may remove some information from the reconstruction obtained by interpolation. In the second case the patch can not be represented as sparsely as the average. This often happens for textured content which cannot be super-resolved accurately with DL-based SR. Therefore, the design decision was made to bypass the SC-based reconstruction in these cases. Finally, the HR image $\boldsymbol{I}_{\mathrm{HR}}$ is obtained by combining all patches.

For the purpose of a valid rate-distortion measurement in ARC it has to be regarded that the rate needed for the low resolution image may be much lower than the rate needed for the full resolution image when both images are coded at the same QP. A straight-forward evaluation and comparison of the rate-distortion cost term

$$J = D + \lambda R$$

would consequently violate the assumption of a linear rate-distortion function at the current operating point of the encoder. For this reason the QP needs to be lowered for the slice containing $\boldsymbol{I}_{\mathrm{LR}}$. Without loosing decoding capability an encoder could try several values for QP and chose the one yielding the best rate-distortion performance. However, this would increase the encoding time dramatically and we chose to set

$$\mathrm{QP}_{\mathrm{LR}} = \mathrm{QP}_{\mathrm{HR}} - 6$$

leading to a bisection of the quantizer step size, as already done in [2]. In this case, the Mean Squared Error (MSE) measured for the low resolution image can be approximated by $\mathrm{MSE}_{\mathrm{LR}} = \frac{1}{4}\mathrm{MSE}_{\mathrm{HR}}$ under the assumption of equally distributed samples within the quantization interval. Consequently, the Sum of Squared Errors (SSE) appears to be close for both low resolution and high resolution since the high resolution image has four times as many pixels as the LR image. Note that these assumptions are only valid as long as the upscaling method is capable of reconstructing almost all frequency components occurring in the coded HR image. This should typically be the case at lower bitrates since transform coefficients representing higher frequencies will be likely be set to zero at low rates, resulting in reconstructed images that appear low pass filtered. Therefore, at lower rates it can be assumed that the upscaled coded LR image and the coded HR image have similar energy spectral densities leading to fair rate-distortion comparisons. At higher rates the distortion introduced by the downsampling will dominate the overall distortion and rate-distortion wise the resolution change will not be able to compete with full resolution coding.

4. SIMULATION SETUP AND RESULTS

All simulations were performed for the 4K test sequences from the JVET test set in the coding scenarios of All Intra (AI) and Random Access (RA). When coding 4K video sequences, the ARC scheme can be operated efficiently since the low resolution image $\boldsymbol{I}_{\mathrm{LR}} \in \mathbb{R}^{1080\times 1920}$, i.e. the tile related to the LR holds an integer number of CTUs in horizontal direction and no padding is required. The parameters for Luma Mapping with Chroma Scaling (LMCS) were calculated based on the HR image only, when ARC was performed. The QP was chosen to lie in the range of $\mathrm{QP} \in \{37, 42, 47, 52\}$ since ARC is assumed to be beneficial at lower bit rates. The anchor rate points were obtained by the VTM-5.0 reference software [15]. The inference of the HR image $\boldsymbol{I}_{\mathrm{HR}}$ was performed using the interpolation filter $\boldsymbol{h}_{\uparrow}$ defined in Sec. 3.2 as a baseline for ARC. Further, the inference was performed by the DL-based SR scheme as machine learning-based improvement. Note that this refinement was only applied to the Y-component of the video because conventional upsampling was assumed to be sufficient for the chroma component due to their low pass characteristics. For the DL-based SR, the patchsize was chosen to be $s_{\mathrm{p}} = 8 \times 8$ and the number of dictionary elements was set to $K = 512$. The penalty λ was set differently for training and testing leading to $\lambda_{\mathrm{train}} = 0.23$ and $\lambda_{\mathrm{test}} = 0.19$. These values were found to result in good performance in a pre-analysis. The dictionaries were trained solving (5) and (6) using the SPAMS software [17]. The training data was sourced from the 91 commonly used images in SR applications.

	All Intra		Random Access	
	$\boldsymbol{h}_{\uparrow}$	DLSR	$\boldsymbol{h}_{\uparrow}$	DLSR
Campfire	$-19.5\,\%$	$-21.0\,\%$	$-12.3\,\%$	$-13.5\,\%$
CatRobot1	$-9.6\,\%$	$-10.7\,\%$	$-5.9\,\%$	$-8.1\,\%$
DaylightRoad2	$-7.2\,\%$	$-8.0\,\%$	$-7.4\,\%$	$-8.1\,\%$
FoodMarket4	$-8.0\,\%$	$-8.2\,\%$	$-12.5\,\%$	$-12.7\,\%$
ParkRunning3	$-16.4\,\%$	$-16.8\,\%$	$-14.7\,\%$	$-15.3\,\%$
Tango2	$-10.0\,\%$	$-10.1\,\%$	$-11.5\,\%$	$-11.7\,\%$
AVG	$-11.8\,\%$	$-12.5\,\%$	$-10.7\,\%$	$-11.6\,\%$

Table 1. BD rate changes with respect to VTM-5.0. The columns labelled with $\boldsymbol{h}_{\uparrow}$ refer to the coding scheme using the interpolation filter $\boldsymbol{h}_{\uparrow}$ defined in Sec. 3.2. The columns labelled with DLSR refer to results obtained when dictionary learning-based super-resolution is used for upscaling.

Table 1 shows the coding results in terms of BD rate changes [18]. It can be observed that by using ARC, the coding results of the plain VTM-5.0 software can be outperformed by far using the ARC scheme. The maximum coding gain is achieved for the Campfire sequence with a rate reduction of $21\,\%$. Moreover, the results show that the DL-based SR helps to further improve the coding gain. On average the SR scheme leads to additional $0.7\,\%$ of rate savings for AI coding and additional $0.9\,\%$ for RA coding respectively. At maximum the additional rate savings gained by DLSR are $2.2\,\%$ for the CatRobot1 sequence in the RA coding configuration. Note that the additional gain is slightly higher for the RA configuration. This shows that the proposed method results in benefits within the coding loop when already upscaled pictures are used for motion compensated prediction. Generally, the achievable gains vary across the test sequences because of the different characteristics of their content. In all cases the DL-based SR scheme outperforms the interpolation filter $\boldsymbol{h}_{\uparrow}$. Therefore, it can be concluded that it makes sense to apply machine learning-based upsampling methods when coding video with adaptive resolution.

5. CONCLUSION

A new approach for ARC based on the concept of uncoded areas and DL-based SR was presented in this paper. The concept of uncoded areas allows for an implementation without many changes to the hybrid video coding scheme in VVC. In summary, the experimental results indicate clearly that applying an ARC makes sense when coding video content at low rates. Furthermore, the SR algorithm can further improve the coding performance in terms of BD rate changes. However, only two different upscaling methods were investigated in this work although the ARC scheme can make use of even more advanced SR algorithms. Therefore, finding an optimum upscaling method could be a major objective for further research as this was not covered by this contribution.

Acknowledgement

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – 407254021

6. REFERENCES

[1] "ISO/IEC 14496-2:2004, information technology – coding of audio-visual objects – part 2: Visual," 2004.

[2] Y. Li, D. Liu, H. Li, L. Li, F. Wu, H. Zhang, and H. Yang, "Convolutional neural network-based block up-sampling for intra frame coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2316–2330, Sep. 2018.

[3] Mariana Afonso; Fan Zhang; Angeliki Katsenou; Dimitris Agrafiotis; David R. Bull, "Low complexity video coding based on spatial resolution adaptation," 2017.

[4] M. Afonso, F. Zhang, and D. R. Bull, "Video compression based on spatio-temporal resolution adaptation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 275–280, Jan 2019.

[5] Anne Aaron, Zhi Li, Megha Manohara, Jan De Cock, and David Ronca, "Per-title encode optimization," Netflix Technology Blog, 2015.

[6] "Advanced video coding for generic audiovisual services," Recommendation ITU-T H.264, Feb. 2014.

[7] B. Bross, W. Han, J. Ohm, G. Sullivan, Y. Wang, and T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS and Last Call)," JCTVC-L1003, Jan. 2013.

[8] Peisong Chen, Tim Hellman, Brian Heng, Wade Wan, and Minhua Zhou, "AHG 8: Adaptive Resolution Change," Tech. Rep. JVET-O0303, JVET, Gothenburg, SE, 2019.

[9] Michael Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer Publishing Company, Incorporated, 1st edition, 2010.

[10] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, Nov 2010.

[11] Roman Zeyde, Michael Elad, and Matan Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*. Springer, 2010, pp. 711–730.

[12] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3467–3478, Aug 2012.

[13] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[14] J. Chen, K. Rapaka, X. Li, V. Seregin, L. Guo, M. Karczewicz, G. V. D. Auwera, J. Sole, X. Wang, C. Tu, Y. Chen, and R. Joshi, "Scalable video coding extension for hevc," in *Data Compression Conference (DCC), 2013*, March 2013, pp. 191–200.

[15] Jianle Chen, Yan Ye, and Seung Hwan Kim, "Algorithm description for Versatile Video Coding and Test Model 5 (VTM 5)," Tech. Rep. JVET-N1002, JVET, Geneva, CH, 2019.

[16] Benjamin Bross, Jianle Chen, and Shan Liu, "Versatile Video Coding (Draft 5)," Tech. Rep. JVET-N1001, JVET, Geneva, CH, 2019.

[17] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[18] Gisle Bjontegaard, "Calculation of average PSNR differences between RD-curves," Tech. Rep. Doc. VCEG-M33, ITU-T SG16/Q6 VCEG, Austin, USA, 2001.