# Dictionary Learning-based Reference Picture Resampling in VVC

Jens Schneider
*Institut für Nachrichtentechnik*
*RWTH Aachen University*
schneider@ient.rwth-aachen.de

Christian Rohlfing
*Institut für Nachrichtentechnik*
*RWTH Aachen University*
rohlfing@ient.rwth-aachen.de

***Abstract*—Versatile Video Coding (VVC) introduces the concept of Reference Picture Resampling (RPR), which allows for a resolution change of the video during decoding, without introducing an additional Intra Random Access Point (IRAP) into the bitstream. When the resolution is increased, an upsampling operation of the reference picture is required in order to apply motion compensated prediction. Conceptually, the upsampling by linear interpolation filters fails to recover frequencies which were lost during downsampling. Yet, the quality of the upsampled reference picture is crucial to the prediction performance. In recent years, machine learning based Super-Resolution (SR) has shown to outperform conventional interpolation filters by far in regard to super-resolving a previously downsampled image. In particular, Dictionary Learning-based Super-Resolution (DLSR) was shown to improve the inter-layer prediction in SHVC [1]. Thus, this paper introduces DLSR to the prediction process in RPR. Further, the approach is experimentally evaluated by an implementation based on the VTM-9.3 reference software. The simulation results show a reduction of the instantaneous bitrate of 0.98 % on average at the same objective quality in terms of PSNR. Moreover, the peak bitrate reduction is measured to 4.74 % for the "Johnny" sequence of the JVET test set.**

***Index Terms*—video compression, reference picture resampling , dictionary learning, versatile video coding**

## I. Introduction

In video coding standards up to High Efficiency Video Coding (HEVC) [2], changing the resolution of a video sequence within the corresponding bitstream required an Intra Random Access Point (IRAP), which is coded at the target resolution. In contrast, Versatile Video Coding (VVC) [3] introduces the concept of Reference Picture Resampling (RPR), which allows to perform a motion-compensated prediction from reference pictures stored at a different resolution than the current picture. Unlike classical coding tools, RPR does not mainly target on improving the Rate-Distortion (RD) performance of the codec, but introduces more flexibility regarding higher level coding concepts:

- A resolution change in an adaptive streaming scenario can be achieved in an open Group of Pictures (GOP) coding configuration [4] instead of introducing another IRAP. On the one hand, the overall bitrate can be reduced, since intra prediction is typically more expensive than inter prediction in terms of RD costs. On the other hand, not every IRAP should be removed from the bitstream, as the Random Access (RA) capability would get lost in that case. However, designing the trade off between overall bitrate and RA capability becomes possible without taking a potential resolution change into consideration by the introduction of RPR.
- In comparison to HEVC, the concept of Scalable Video Coding (SVC) is not to be specified as an extension to VVC, as it's functionality can be approached by combining layered coding and RPR [5]. Due to the ability to predict from pictures at lower resolution, pictures from a layer, representing a lower resolution, can be made available for prediction by placing them into the reference picture lists of the current picture.
- To be mentioned lastly, RPR could be applicable in a video-telephony scenario with varying channel capacity. In this case, downsampling and upsampling the transmitted video is highly beneficial in order to approximate the convex hull of the true RD curve, but introducing intra pictures would result in instantaneous bitrate peaks. Apparently, those peaks are non desirable, when the available bitrate is limited. Thus, RPR offers a suitable solution to avoid high instantaneous bitrates, while retaining the ability to change the resolution of the transmitted video sequence.

Since a resolution change is required for RPR, a downsampling and a respective upsampling method were specified in VVC. Both, downsampling and upsampling, are implemented by linear filtering operations combined with either dropping samples or inserting zeros, respectively. However, recent research results showed that machine-learning-based Super-Resolution (SR) leads to improved performance, when applied to video coding systems [6], [7]. Furthermore, the potential of Dictionary Learning-based Super-Resolution (DLSR) for predicting pictures of a video sequence at a higher resolution was demonstrated in the context of SHVC [1]. Therefore, the upsampling method of RPR is replaced by DLSR and experimentally evaluated in this paper.

The rest of the paper is organized as follows: The fundamentals of RPR and DLSR are provided in Section II-A and II-B, respectively. In section III, the implementation and performance measures for DLSR-based RPR are addressed. The experimental setup and corresponding results in terms of luma Bjøntegaard Delta (BD) rate changes are provided

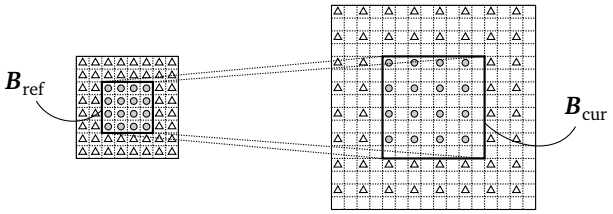


Fig. 1. Reference Picture Resampling on the block level.

in section IV. Finally, section V concludes the paper.

## II. Fundamentals

### A. Reference Picture Resampling

As mentioned above, the concept of RPR requires a resolution change. Both, applying and testing the RPR functionality, requires certain frames of the input video sequence to be downsampled. This downsampling operation of entire frames is not specified in VVC, as it affects only the encoding process but not the structure of the bitstream. However, for the reference implementation in VTM-9.3 [8], downsampling by a factor of 2 is performed with the anti-aliasing filter

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

which is applied both horizontally and vertically.

From the specification perspective, RPR is operated on a per-block level as part of the motion compensation in VVC. In particular, motion compensation and resampling of the reference block are performed together. To this end, each pixel of the prediction signal for the current block $\boldsymbol{B}_{\text{cur}}$ is calculated by the inner product between the applicable sub-pel interpolation filter and the samples from the reference block $\boldsymbol{B}_{\text{ref}}$. Fig. 1 shows the exemplary case of a reference picture possessing half the resolution of the current picture and zero motion between the blocks $\boldsymbol{B}_{\text{ref}}$ and $\boldsymbol{B}_{\text{cur}}$. In this case, the pixels at zero-phase positions (∘) are copied to the prediction signal and missing pixels are calculated by interpolation filtering. The filtering operation requires pixels lying beyond the block boundaries (△) as shown in the figure. The size of the boundary extension is defined by the length of the interpolation filter. Note that the number of 2 additional lines and columns was only chosen for illustration purposes.

The applicable filter depends on both, the sub-pel accuracy of the current motion vector and the position of the current pixel in the current block. However, from a signal processing point of view, this implementation is identical[1] to performing the motion compensation first and upsampling the corresponding block in a second step. In this case, the interpolation filter can be identified as

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

Thus, the combination of anti-aliasing filter and interpolation filter is the same as in the reference software of SHVC, namely SHM [9]. Generally, the reference picture could also be available at the higher resolution, when downsampling of the video is performed. In this case, conventional downsampling is considered to be appropriate, as it retains the entire frequency range of interest.

[1] besides rounding effects

### B. Dictionary Learning-based Super-Resolution

DLSR relies on the assumption that a vectorized natural image patch $\boldsymbol{x} \in \mathbb{R}^{s_{\text{p}}^2}$ can be represented sparsely in a trained dictionary $\boldsymbol{D} \in \mathbb{R}^{s_{\text{p}}^2 \times K}$, where the square patch has dimensions of $s_{\text{p}} \times s_{\text{p}}$, and the dictionary contains $K$ atoms. Then, the representation up to a model error $\boldsymbol{\varepsilon}$ reads

$$\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha} + \boldsymbol{\varepsilon}\,, \tag{1}$$

with $\boldsymbol{\alpha} \in \mathbb{R}^K$ denoting a sparse coefficient vector. Typically, the patches are extracted in an overlapping manner, in order to avoid blocking artefacts, when an entire image is processed. Moreover, the extracted patches are often centered (i.e. their mean values are subtracted) and normalized to unit $\ell_2$-norm in order to ensure the stability of both Dictionary Learning (DL) and Sparse Coding (SC) algorithms. When the dictionary is known, the sparse coefficient vector $\boldsymbol{\alpha}$, which represents an image patch, can be found by the solution to the SC problem

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

where $\lambda$ denotes a penalty controlling the level of sparsity in the solution. Analogously, a dictionary can be trained from natural image patches $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N]$ by solving the DL optimization task

$$\boldsymbol{D} \leftarrow \arg\min_{\boldsymbol{D}} \sum_{j=1}^{N} \frac{1}{2}\|\boldsymbol{x}_j - \boldsymbol{D}\boldsymbol{\alpha}_j\|_2^2 + \lambda\|\boldsymbol{\alpha}_j\|_1. \tag{3}$$

Generally, the SC problem (2) can be approached by e.g. the LARS algorithm and DL (3) can be performed by the Online Dictionary Learning (ODL) algorithm. The implementations of this work were based on the SPAMs software, which implements DL/SC algorithms [10].

Unlike the simple representation of image patches in a single dictionary, DLSR requires two dictionaries $\boldsymbol{D}_{\downarrow}$ and $\boldsymbol{D}_{\uparrow}$ with coupled sparsity. Thereby, coupled sparsity refers to the property that the sparse coefficient vector $\boldsymbol{\alpha}$ is shared among both dictionaries for representing either the low resolution image or the high resolution image, respectively. Thus, once the sparse representation of a low resolution image patch in $\boldsymbol{D}_{\downarrow}$ is found, the reconstruction in the high resolution dictionary $\boldsymbol{D}_{\uparrow}$ using the same coefficients $\boldsymbol{\alpha}$ yields an estimate for the corresponding high resolution image patch. In order to ensure that the dictionaries indeed possess the property of coupled sparsity in a least squares sense, the high resolution dictionary is trained as

$$\begin{aligned} \boldsymbol{D}_{\uparrow} &\leftarrow \arg\min_{\boldsymbol{D}_{\uparrow}} \frac{1}{2}\left\|\boldsymbol{D}_{\uparrow}\boldsymbol{A} - \boldsymbol{X}\right\|_2^2 \\ \Leftrightarrow \boldsymbol{D}_{\uparrow} &= \boldsymbol{X}\boldsymbol{A}^{+}\,, \end{aligned} \tag{4}$$

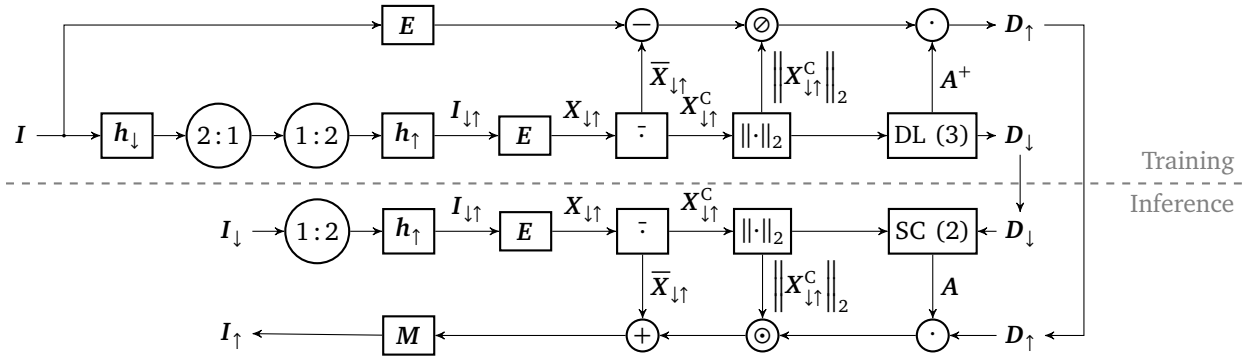


Fig. 2. Training and inference for DLSR. Variables correspond to training set and test set, even if they are denoted by the same symbol. The operators $\odot$ and $\oslash$ refer to element-wise multiplication and division, respectively.

with $\boldsymbol{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_N]$ being the sparse codes found in $\boldsymbol{D}_\downarrow$ and $\boldsymbol{A}^+$ being their pseudoinverse.

Fig. 2 depicts the training and inference stage of DLSR as a detailed block diagram for a scaling factor of $s_{\downarrow\uparrow} = 2$ between low resolution and high resolution. The inference stage of the figure reveals that the low resolution image $\boldsymbol{I}_\downarrow$ is upsampled and interpolated by $\boldsymbol{h}_\uparrow$ prior to the rest of the processing chain. This prior upsampling is typically performed in DLSR in order to process all patches at the same resolution [11]. Obviously, the training is designed analogously and the anti-aliasing and interpolation filters $\boldsymbol{h}_\downarrow$ and $\boldsymbol{h}_\uparrow$ define the relationship between the features $\boldsymbol{I}_{\downarrow\uparrow}$ and the labels $\boldsymbol{I}$ in the training process. Furthermore, the block diagram entails the patch extraction and combination denoted by the operators $\boldsymbol{E}$ and $\boldsymbol{M}$, and the preprocessing operations of centering ($\bar{\cdot}$) and normalization ($\|\cdot\|_2$). Conceptually, the patch combination operator $\boldsymbol{M}$ could be optimized with respect to the application at hand, but is typically chosen to average multiple estimates for a singe pixel value in DLSR [11], [12]. Hence, the operator $\boldsymbol{M}$ is considered as an averaging operator in this paper. In summary, the DLSR scheme provides an approximation of the function which is inverse to filtering an image with the anti-aliasing filter $\boldsymbol{h}_\downarrow$ followed by downsampling.

## III. DLSR-based RPR and Performance Measures

When introducing DLSR to RPR, the most intuitive approach would be to operate DLSR as a refinement to the prediction signal at the decoder side. The initial upsampling (cf. Fig. 2) could be omitted in this case, as the upsampling and interpolation would be performed as part of the prediction process already. A benefit of this strategy would be that DLSR is only applied for blocks, which are actually predicted from a lower resolution. However, from the encoder perspective, this strategy is not practical in terms of computational complexity, as DLSR would need to be performed many times as part of the motion estimation in the RD optimization. Therefore, we chose to make the prediction signal obtained by DLSR available by prepending an additional reference picture to the reference picture lists of the current picture, instead. This additional reference is generated by applying DLSR to the temporally nearest frame at lower resolution. Since the additional reference appears at the same resolution as the current picture, it can be used for prediction directly without any further processing. Consequently, the resulting computational complexity of the motion estimation is the same as in coding scenarios with constant resolution.

As mentioned in section II-A, a resampling operation of the input video needs to be performed in order to evaluate the performance of different upsampling methods for RPR. According to the testing conditions for RPR [13], the video sequence is coded in a low delay B configuration and a resampling operation should take place approximately every 0.5 s such that it appears at the beginning of a GOP. Fig. 3 depicts the resampling scheme of the test configuration. The positions of a change to the higher resolution, e.g. $2n + 1$, are referred to as Upsampling points (UPs) in the following. Because of the varying resolution of the reconstructed frames, the conventional quality assessment via the PSNR between the original and reconstructed frames becomes difficult. Hence, it was proposed to either measure the PSNR in the low resolution domain, denoted as PSNR-1, or to upsample the reconstructed frames and measure the PSNR on the high resolution images, referred to as PSNR-2 [13]. Eventually, both measures could be used for the calculation of BD rate changes [14]. However, PSNR-2 might lead to misleading results: It is highly dependent on the upsampling method used at the decoder which is not specified and can be considered as post processing. Therefore, PSNR-2 was excluded from the evaluation in this paper.

Furthermore, both measures, PSNR-1 and PSNR-2, do not target the relative changes of the instantaneous bitrate. To take these changes into account, we introduce a third measure, PSNR-3, which is measured only on frames at UPs and averaged over these frames. Consequently, BD-rate statistics based on the PSNR-3 do not report overall rate changes but the changes in the instantaneous bitrate at UPs.

## IV. Experimental Setup and Results

Simulations according to the testing conditions for RPR [13] in the configuration with a scaling factor of $s_{\downarrow\uparrow} = 2$ were performed for the sequences from the JVET test set. More precisely, the sequences were coded in a low delay B coding scenario with QP $\in \{22, 27, 32, 37\}$. The anchor

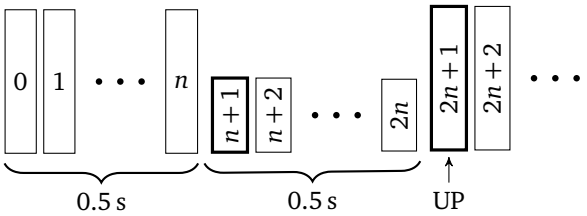


Fig. 3. Frame resolutions in testing conditions for RPR. Points, at which the resolution is increased are referred to as upsampling points (UP).

rate points were defined by the VTM-9.3 reference software, which further served as basis for the implementation of the proposed approach. The DLSR scheme was trained and parameterized as follows: The dictionaries were trained with $K = 512$ atoms at a patch size of $s_p \times s_p = 8 \times 8$. Further, the training was performed using the 91 training images commonly used in DL [11], [15]. In both training and inference stage, the sparse coding penalty was set to $\lambda = 0.01$, which was evaluated as appropriate in a preliminary experiment. Moreover, the patches were extracted with a step size of $s_s = 2$ such that a new patch starts at a zero-phase pixel position in the initial upsampled image [11].

Table I reports the luma BD rate changes measured for PSNR-1 and PSNR-3, respectively. It can be observed that the overall bitrate at the same PSNR-1 can only be reduced significantly for the sequences from the class E sequence. For the other classes, RPR including DLSR performs on par with the anchor. However, the results measured for PSNR-3 provide a clear indication that DLSR improves the coding performance of RPR regarding the instantaneous bitrate also for other sequences such that on average 0.98 % of bitrate savings are measured. The highest coding gains are achieved for the class E sequences. This behavior can be explained by the static high frequency background in these sequences, as DLSR is known to perform best on high frequency structures such as sharp edges [16]. Moreover, for static areas in the video sequences, the prediction process in RPR becomes identical to the inter-layer prediction process in SHVC, as the high resolution content needs to be predicted from a low resolution representation at the same spatial position. Therefore, the higher gains for the class E sequences match the expectation, since DLSR is known to outperform conventional interpolation filters in SHVC [1].

Unlike the significant gain for the class E sequences, for some sequences of the other classes slight losses are observed, especially for PSNR-1. This is explainable by DLSR not outperforming interpolation clearly on rather low frequency images. For instance, the “FoodMarket4” and the “Tango2” sequence feature smooth areas and less sharp edges. In consequence, the additional reference does not provide a significantly better prediction signal than RPR, but the bitrate is slightly increased due to the increased number of reference picture indices. However, the losses are negligible except for the “SlideShow” sequence. For this sequence, the loss was found to be caused by a worse performance of DLSR-based RPR at lower bit rates. Therefore, it could be explainable by the severe influence of coding artefacts on the DLSR scheme at lower rates. In summary, the results indicate the potential for DLSR to improve the prediction of RPR in VVC, especially with respect to the instantaneous bitrate at UPs.

TABLE I
LUMA BD RATE STATISTICS OF DICTIONARY LEARNING-BASED RPR IN %.

| Class | Sequence | PSNR-1 | PSNR-3 |
|---|---|---|---|
| A1 | Campfire | 0.04 | −0.01 |
| A1 | FoodMarket4 | 0.05 | 0.13 |
| A1 | Tango2 | 0.03 | 0.09 |
| A2 | CatRobot1 | −0.13 | −1.43 |
| A2 | DaylightRoad2 | 0.08 | −0.23 |
| A2 | ParkRunning3 | 0.02 | −0.12 |
| A | AVG | 0.02 | −0.26 |
| B | BQTerrace | −0.27 | −1.62 |
| B | BasketballDrive | −0.03 | −0.24 |
| B | Cactus | −0.29 | −1.81 |
| B | MarketPlace | 0.08 | −0.1 |
| B | RitualDance | −0.05 | −0.3 |
| B | AVG | −0.11 | −0.81 |
| C | BQMall | 0 | −0.56 |
| C | BasketballDrill | −0.17 | −1.9 |
| C | PartyScene | 0.03 | −0.31 |
| C | RaceHorsesL | 0.15 | 0.21 |
| C | AVG | 0 | −0.64 |
| D | BQSquare | 0.16 | −0.16 |
| D | BasketballPass | 0.06 | −0.11 |
| D | BlowingBubbles | −0.02 | −0.33 |
| D | RaceHorsesM | 0.11 | 0.06 |
| D | AVG | 0.08 | −0.14 |
| E | FourPeople | −0.77 | −4.06 |
| E | Johnny | −1.08 | −4.74 |
| E | KristenAndSara | −1.18 | −4.42 |
| E | AVG | −1.01 | −4.41 |
| F | ArenaOfValor | −0.16 | −1.24 |
| F | BasketballDrillText | 0.02 | −1.45 |
| F | SlideEditing | −0.37 | −1.31 |
| F | SlideShow | 0.13 | 0.5 |
| F | AVG | −0.10 | −0.88 |
| | AVG | −0.14 | −0.98 |

## V. CONCLUSION

The concept of DLSR was introduced to RPR and evaluated in comparison to the VTM-9.3 reference software in this paper. The experimental results show clearly that DLSR outperforms conventional interpolation filters, when applied in the prediction process of RPR. In particular, the instantaneous bitrate can be reduced significantly, while maintaining the quality of the reconstructed video. Conceptually, also other SR methods could be used for the prediction and the proposed implementation is not limited to DLSR. However, the investigation of e.g. SR algorithms relying on Convolutional Neural Networks (CNNs) is left open for further research. In conclusion, a general scheme for introducing advanced SR methods to RPR is provided alongside with promising coding results achieved by DLSR.

## ACKNOWLEDGMENT

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

## REFERENCES


[1] J. Schneider, J. Sauer, and M. Wien, "Dictionary Learning based High Frequency Inter-Layer prediction for Scalable HEVC," in *Proc. of IEEE Visual Communications and Image Processing VCIP '17*. St. Petersburg, USA: IEEE, Piscataway, Dec. 2017.

[2] *High efficiency video coding*, Recommendation ITU-T H.265, Std. Recommendation ITU-T H.265.

[3] *Versatile video coding*, Recommendation ITU-T H.266, Std. Recommendation ITU-T H.266.

[4] D. Gibellino, "Versatile video coding hits major milestone," Medium Blog, Diego Gibellino, 2019. [Online]. Available: https://medium.com/@gibellino/versatile-video-coding-hits-major-milestone-baeb13c8960a

[5] Y.-K. Wang, F. Hendry, and J. Chen, "AHG 8: Scalability for VVC - general," JVET, Gothenburg, SE, Tech. Rep. JVET-O0135, 2019.

[6] J. Schneider, J. Sauer, and C. Rohlfing, "Adaptive resolution change using uncoded areas and dictionary learning-based super-resolution in versatile video coding," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 2203–2207.

[7] K. Fischer, C. Herglotz, and A. Kaup, "On versatile video coding at uhd with machine-learning-based super-resolution," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, 2020, pp. 1–6.

[8] J. Chen, Y. Ye, and S. H. Kim, "Algorithm description for versatile video coding and test model 9 (vtm 9)," JVET, Tech. Rep. JVET-R2002.

[9] J. Chen, J. Boyce, Y. Ye, M. Hannuksela, and G. Barroux, "SHVC test model 11 (SHM 11) introduction and encoder description," JCTVC, 22nd Meeting, Geneva, Tech. Rep. JCTVC-V1007, oct 2015.

[10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th annual international conference on machine learning*. ACM, 2009, pp. 689–696.

[11] R. Zeyde, M. Elad, and M. 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] J. Luo and V. Seregin, "JVET functionality confirmation test condition for reference picture resampling," JVET, Brussels, BE, Tech. Rep. JVET-Q2015, 2020.

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

[15] 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.

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