UNIVERSIDAD NACIONAL DE COLOMBIA

Sparse In-loop Video Coding Restoration Method

Carlos Alberto Salazar Herrera

National University of Colombia
Faculty of Mines
Department of Computing and Decision Sciences
Medellin, Colombia

2023

Sparse In-loop Video Coding Restoration Method

Carlos Alberto Salazar Herrera

This dissertation is submitted in partial fulfillment for the degree of:
Doctor of Engineering - Systems and Informatics

Supervisor:
John W. Branch, Ph.D
Universidad Nacional de Colombia

Co-supervisor:
Maria P. Trujillo, Ph.D
Universidad del Valle

National University of Colombia
Faculty of Mines
Department of Computing and Decision Sciences
Medellin, Colombia
2023

Declaration of Authorship

Candidate's declaration

I, Carlos Salazar, hereby declare that I am the author of this thesis, that I have consulted all references cited herein, that the work of which this thesis is a record has been done by me, and that it has not been previously accepted for a higher degree.

Signed Date

Supervisors' declaration

We, Jhon W. Branch and Maria P. Trujillo, hereby declare that we are the supervisor of the candidate, and that the conditions of the relevant Ordinance and Regulations have been fulfilled.

Signed Date
Universidad Nacional de Colombia

Signed Date
Universidad del Valle

To my lovely Wife and Daughter.

"Be strong and courageous. Do not be frightened, and do not be dismayed, for the Lord your God is with you wherever you go"

- Joshua 1:9

Acknowledgment

I thank Professor John Willian Branch for the opportunity he gave me to start this adventure a couple of years ago and for the guidance, support, and motivation throughout this process. In addition, since my undergraduate, I thank Professor Maria Patricia for being a guide and an engine to achieve new challenges and for teaching me how exciting and wonderful the world of video compression is. In the same way, I thank the National University of Colombia and the entire academic community that was part of my process during these years.

Abstract

Video in-loop restoration methods have traction much attention across the standardization groups for future video codecs AV2 [1] and VVC [2]. Primarily, because of potential benefits to compensate the artifacts generated during super-resolution scenarios and the effect of quantization process. Thus, new sophisticated learned-based algorithms have been proposed, in recent years, surpassing the classical switchable filter implementation in objective quality rate. However, CNN-based approaches requirements on computational cost and decoder complexity are still challenging.

Therefore, we propose a low-complex learned-based method that leverages the solid and consistent sparse representation theory to exploit the spatial redundancy of frames. Our approach models the decoding residual, the distance between each reference and the respective decoded frame. Furthermore, the proposed methods shrink the support of the sparse vector to two in order to control the restoration signal information. In addition, our method uses the Discrete Cosine Transform (DCT) orthogonal basis as a dictionary to exploit the statistical correlation between nonzero coefficients and the quantization level. Finally, we leverage the official and public available AV2 raw video dataset to compare our performance against the anchor AV2 codec through three objective visual quality metrics. The validation protocol includes benchmark data sets for the anchor and the restoration-enabled configurations. Our experimental results show a consistent restoration using sparse representation as well as an effective mechanism for sharing nonzero coefficients leveraging a Gaussian correlation. The experimental evaluation showed that our method has a 1%-2% gain regarding AV2, using SSIM and VMAF under similar bitrate conditions.

Key Words: AV2, HEVC, VVC, QP, PCA, Sparse, Dictionary, PSNR.

[1] https://aomedia.org/

[2] Versatile Video Coding https://jvet.hhi.fraunhofer.de/

Título: Método para la restauración de video en el bucle del proceso de compresión

Resumen:
Los métodos de restauración de vídeo en bucle han venido incrementando el intereste por parte los grupos de estandarización para los futuros códecs de vídeo AV2 y VVC. Esto principalmente debido a a los beneficios potenciales para compensar efectos no deseados en el video producidos durante los procesos de super-resolución y cuantización. Así, en los últimos años se han propuesto nuevos y sofisticados algoritmos basados en aprendizaje, que superan a la clásica implementación de filtros conmutables en cuanto a tasa de calidad objetiva. Sin embargo, los requisitos de los enfoques basados en CNN en cuanto a coste computacional y complejidad del descodificador siguen siendo un desafio. Por ello, proponemos un método de baja complejidad basado en aprendizaje, que aprovecha la sólida y consistente teoría de la representación dispersa para explotar la redundancia espacial de los fotogramas que componen un video. Nuestro enfoque modela el residuo de descodificación, la distancia entre cada referencia y el respectivo fotograma descodificado. Además, el método propuesto reduce el soporte del vector disperso a dos para controlar la información de la señal de restauración. Por otra parte, nuestro método utiliza la base ortogonal de la transformada discreta de coseno (DCT) como diccionario para explotar la correlación estadística entre los coeficientes distintos de cero y el nivel de cuantificación. Por último, aprovechamos el conjunto de datos de vídeo de AV2, oficial y público, para comparar nuestro rendimiento con el códec AV2 de referencia, mediante tres métricas objetivas de calidad visual.El protocolo de validación incluye conjuntos de datos de referencia para las funciones de anclaje y restauración. Nuestros resultados experimentales muestran una restauración coherente utilizando una representación dispersa así como un mecanismo eficaz para compartir coeficientes distintos de cero aprovechando una correlación gaussiana. La evaluación experimental mostró que nuestro método tiene una ganancia del 1%-2% con respecto a AV2, utilizando SSIM y VMAF en condiciones de bitrate similares.

Palabras claves: AV2, HEVC, VVC, QP, PCA, Sparse, Dictionary, PSNR.

Content

List of Figures

List of Tables

1 Introduction

This thesis focuses on the in-loop restoration techniques, particularly on reducing complexity and increasing the objective video quality (PSNR, VMAF [1], and SSIM [46]) while maintaining the bit rate. Most of the work described herein was done with the collaboration of Elemental Amazon Web Services (AWS), who provides expert guidance and infrastructure to perform extensive testing on the reference video codec AV2. Therefore, the research's target is to design an in-loop restoration method for AV2 open source codec. This chapter provides the motivation for working on in-loop restoration, the research problem and the aims guided this research. Then, I summarize the main contribution and discuss the organization of the thesis.

1.1 In-loop restoration

In-loop video restoration is a topic that has gained the attention of the leading video codec standardization groups. It is because frame denoising and deblurring significantly impact visual quality and compression efficiency, which are the main topics of AV2 and VVC. Nevertheless, the strategy to tackle restoration problems could be more precise, especially under computational complexity restriction and objective quality metrics, the last because PSNR is not the more human perception correlated metric. As expected, the convolutional network has landed to demonstrate its efficiency in image processing tasks. Outstanding examples are the filters introduced by Ding *et al.* [9] and Kong *et al.* [28]. Both approaches leverage learning of distorted and clean images, assisted by the quantization compression level, to obtain superior restoration performance in terms of PSNR BD-rate against classical algorithms. However, it has tremendous implementation complexity, especially on the decoder side($\sim 30 \times x$). On the other hand, well-known filters, such as the separable Wiener [43] and sample adaptive filter (SAO) [8], report significant efficiency improvements (3% BD-Rate [41] gain) despite the processing time due to the number of recursive operations at the

[1] https://github.com/Netflix/vmaf

restoration unit basis.

That said, three main challenges for in-loop restoration are identified: 1) Increasing (BD-Rate) by relying on more human-visual correlated quality metrics, such as VMAF. 2) Holding reasonable complexity (<10x) compared with the anchor video codec, and 3) Maintaining a low bitrate for restoration signal information.

1.2 Problem statement

Consequently, this research focuses on the three main in-loop restoration problem: 1) Increasing objective visual quality (SSIM, PSNR, VMAF) against the anchor AV2 codec, 2) Maintaining a proper restoration signaling bitrate, and 3) Reducing the complexity in both, the encoder and decoder, sides.

1.3 Aims

Propose a video compression in-loop restoration method, using sparse learned-based techniques, that increases objective visual quality while maintains a reasonable efficiency and complexity.

The proposed method requires to:

- Build a videos data-set for evaluating video compression in-loop restoration approaches that guarantees properly standards.
- Deploy an in-loop restoration method for AV2 video compression codec.
- Design an assessment protocol for evaluating the in-loop restoration method against anchor AV2.

1.4 Thesis contributions

The main contributions of this research are:

- A video compression in-loop restoration method that increases objective visual quality while maintaining reasonable efficiency and complexity. The method relies on the

sparse representation of the decoding residual (DR) and the prediction of the nonzero sparse coefficients through a Gaussian estimators adjusted accordingly to the selected QP. These reduces the restoration signaling bitrate between the encoder and decoder.

- A video data set for evaluating video compression in-loop restoration approach using three configurations: 1) Restoration-disable, 2) Restoration-enable, and 3) Sparse-restoration enable. All three configurations are processed using six QP levels and generating PSNR, VMAF, and SSIM as objective video quality performance metrics.

- An in-loop restoration prototype integrated to AV2 reference video compression codec and public accessible [2].

- An assessment protocol for evaluating in-loop restoration methods based on sparse representation and coefficients estimation. The protocol follows the common standard conditions defined by AOMedia.

1.5 Thesis organization

The thesis is structure as follows: we present in chapter 2 the current methods for in-loop restoration in the most recent AOM Video codec standard (AV1), which is the reference for developing the future video codec (AV2). This chapter illustrates the transform and quantization blocks beside the algorithms that compound the switchable restoration filter. Next, we describe in chapter 3 the basis of the sparse representation theory with particular attention to techniques for defining and learning the dictionary considering the problem addressed in this thesis. Following in chapter 4, we present a theoretical analysis and propose a framework for an in-loop restoration method leveraged by three main contributions: 1) Sparse decoded residual, 2) Sparse coefficients estimator and 3) Sparse position estimator. All contributions are QP-awareness and design to add a reasonable signal information and complexity. Next, we define our experimental and validation methodology in chapter 5 where the performance of the proposed method is contrast against the anchor AV2 and two CNN-based restoration strategies. Finally, in chapter 6, we present the conclusions and future works.

[2] https://github.com/casalazarh/sparse-in-loop-restoration

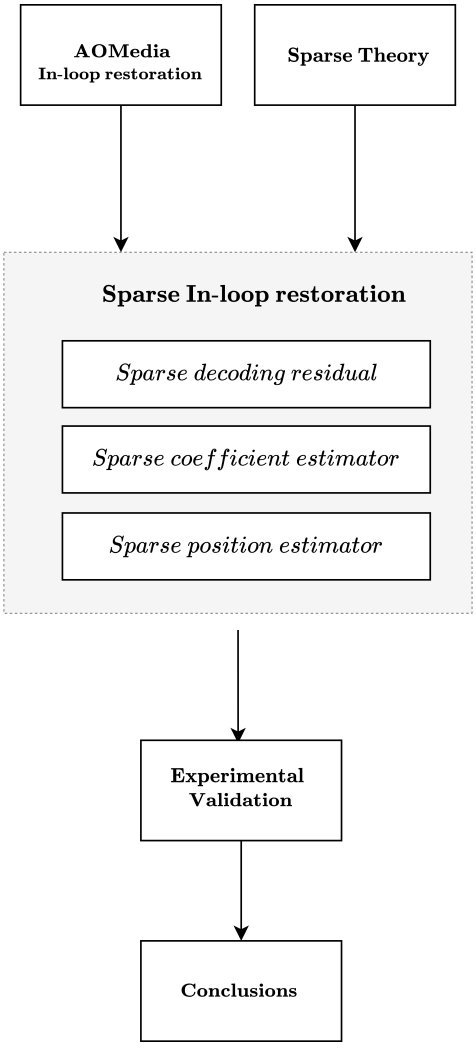

Figure 1-1: Thesis organization.

2 AOMedia In-loop restoration tools

The Alliance for Open Media (AOMedia) is a collaboration between leading tech companies aiming to make video compression standards royalty-free and widely adopted. Companies such as Amazon, Netflix, Google, and Apple belong to AOMedia. In 2018 the alliance officially released the AV1 video codec. Since then, the global media industry has embraced it while addressing the challenge of hardware support on various end-user devices, including phones, tablets, and Smart-TV. In the last two years, the addition of chips supporting AV1 has grown exponentially, accelerating its adoption (28 % YoY), as depicted in figure **2-1**. It starts establishing a potential ecosystem to ensure the success of adopting the coming AV2 codec. Therefore academics and industry are actively working on new approaches to enhance the baseline video codec (AV2), illustrated in figure **2-2**, at the distinct functional blocks while maintaining a reasonable compatibility and complexity.

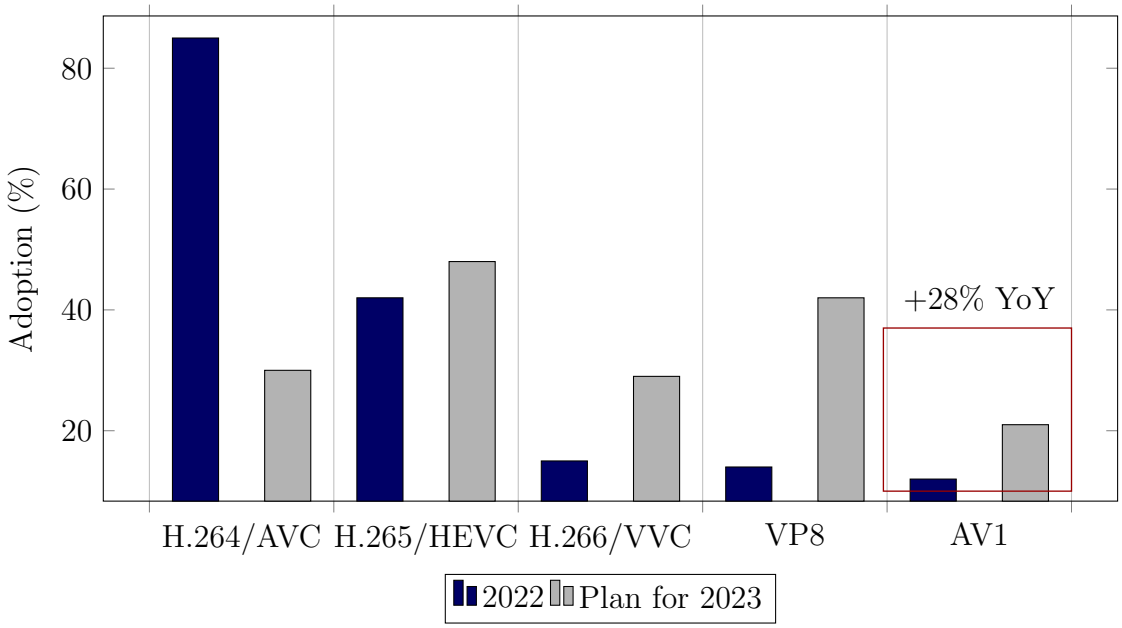

Figure 2-1: Video codec adoption in 2022 and plan for 2023 (source : Bitmovin 2022 report).

We leave the general video architecture out of the scope of the chapter and encourage read-

ers to obtain further details in [21]. Instead, we concentrate on explaining the details of the transform and quantization of blocks followed by the post-processing stage, including deblocking, constrained directional enhancement, and in-loop filters. Those topics provide enough background for the discussion in Chapter 4 where our framework is presented.

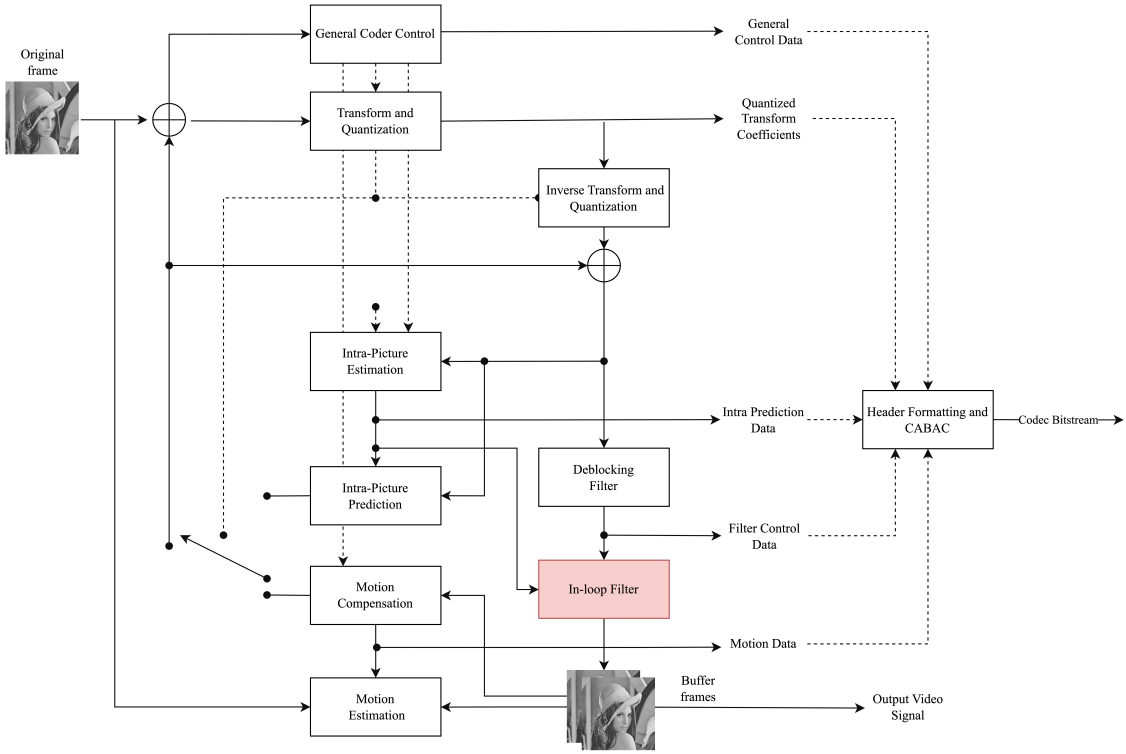

Figure 2-2: General AOMedia Reference Video Codec Architecture.

2.1 Transform & Quantization

Expanding a signal into another space, such as the frequency domain, permits eliminating redundant information. For instance, in images case, the frequency domain allows us to separate DC and AC components to eventually truncate lower frequency that does not impact the visual human perception. AV1 extends the definition of transformation operator –of its predecessor VP9 [34]– in 1) the size of blocks become dynamic and 2) the size of transform kernels from 2 to 4. As following, a description of those transformation.

2.1.1 Transform Block Size

AV1 supports square block sizes from $4 \times 4$to a max of 64×64, and includes rectangular sizes $N \times N/2$, $N/2 \times N$, $N \times N/4$, and $N/4 \times N$. AV1 improved the transform coding efficiency by capturing localized stationary regions for all inter-coded blocks throughout the recursive transform block. The initial transform block size is the same as the coding block size, unless it is more significant than 64×64. In that case, the block size will be 64×64. The recursive transform can go up to 2 levels for the luma component. For the intra-coded block, AV1 inherits the uniform transform block size approach. Therefore, the maximum transform block size matches the inter-coding block size and can go up to 2 levels down for the luma component. The chroma components tend to have much less variation in the statistics. For this reason, the transform block is set to use the most extensive available size.

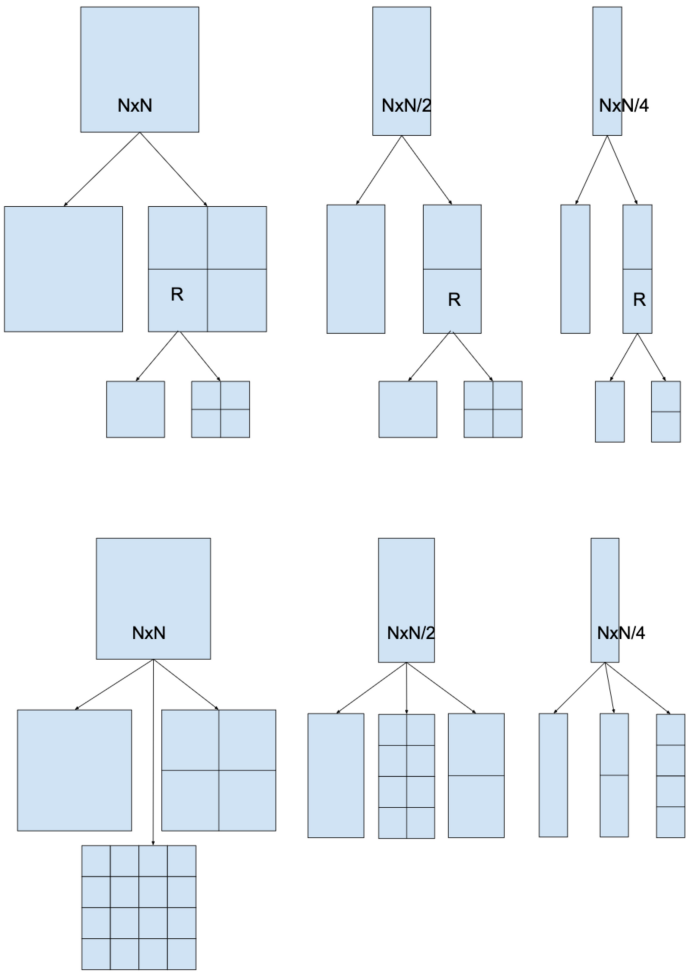

Figure 2-3: Transform block partition for Inter and Intra mode.

2.1.2 Transform Kernels

AV1 takes the exact definition of the VP9 Kernels; the difference is that it allows each transform block to apply its transform kernel independently. It also combines two extra 2-D separable transform kernels to give a total of four 1-D kernels: DCT [2], ADST [22], flipped ADST (FLIPADST), and identity transform (IDTX), that four 1-D combinations that result in 16 2-D transform kernels. The criteria for selecting the kernels rely on statistics and accommodate various boundary conditions. Each kernel has intrinsic advantages; i.e., DCT approximates the optimal linear transform, ADST and FLIPADST are naturally suitable for coding some intra-prediction residuals, and IDTX fits better in cases where sharp transitions are contained in a block, and neither DCT nor ADST is effective. Furthermore, the IDTX, combined with other 1-D transforms, provides the 1-D transforms themselves, therefore allowing for better compression of horizontal and vertical patterns in the residual. The inverse transform is well known for its computational cost. However, a notable reduction is achieved using a butterfly structure for multiplication operations over simple matrix multiplication. Furthermore, due to its efficiency with large transform block sizes, it is used for a block size of 8×8 and above. When trying to adjust those boundary conditions, when those are less pronounced, all the sinusoidal transforms largely converge for large transform block sizes. That is why DCT and IDTX are used for blocks with dimensions bigger or equal to 32×32.

2.1.3 Quantization

Once the coefficients of a specific transform are obtained, they are quantized into discrete steps which depend on the selected QP parameter and vary from 0 to 255, where zero represents loss mode. The quantization block relies on two main concepts:

- **Quantization Step size**: this leverage the human-visual system tolerance to frequency distortion. AV1 supports 15 sets of pre-defined quantization weighting matrices, where the quantization step size for each frequency component is further scaled differently.

- **Quantization Parameter Modulation**: AV1 defines a hierarchical mechanism to represent the coefficient for both AC and DC. Its starts with QP_{base} assigned at the frame level. Next, an offset value is sent throughout the header ($\Delta QP(p, b)$), associating the color planes. $p \in Y, U, V$ denotes the plane and $b \in DC, AC$ represents the DC/AC transform coefficients. AV1 permits QP offset headers at the

superblock and coding block levels to compensate for some frames' rate distortion. Therefore, the effective QP for AC coefficients in a coding block, QP_{cb}, is given by $QP_{cb} = clip(QPframe + \Delta QP_{sb} + \Delta QP_{seg}, 1, 255)$, where ΔQP_{sb} and ΔQP_{seg} are the QP offsets from the superblock and the segment, respectively. The clip function ensures it stays within a valid range. The QP cannot change from a nonzero value to zero since zero is reserved for lossless coding (Table **2-1**).

	AC	DC
Y	QP_{base}	$QP_{base} + \Delta QP_{Y,DC}$
U	$QP_{base} + \Delta QP_{U,AC}$	$QP_{base} + \Delta QP_{U,DC}$
V	$QP_{base} + \Delta QP_{V,AC}$	$QP_{base} + \Delta QP_{V,DC}$

Table 2-1: Hierarchical QP mechanism.

2.2 Post-processing filters

The reference codec AV1 supports three post-processing filters that are enabled independently. The post-processing stage starts with the blocks resulting from the addition between the prediction and the residual. Then, the blocks are ready to fill in the Inter-prediction buffer and pass to the display picture module. Before that, the post-processing filter might be applied with two main objectives: 1) Improve the Inter-prediction due to the most accurate reconstruction frame in the buffer and reduce overall bitrate. 2) Increase visual output quality. That is a typical case for super-resolution, where before post-processing filters, there is an upscaling operation that naturally truncates image details. Figure **2-4** presents the high-level workflow of the post-processing stage. Each filter is also further detailed next.

2.2.1 Deblocking filter

During the encoding process, a residual frame, between a reference and a predicted frames, is split into transform blocks of different sizes: $4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64$. Each block is computed using the discrete cosine transform (DCT), or the asymmetric discrete sine transform (ADST) to eliminate spatial correlation. Then, the resulting coefficients are quantized into N levels determined by the QP parameter. The signal information regarding quantization is entropy coded and sent to the decoder. The decoder performs the inverse

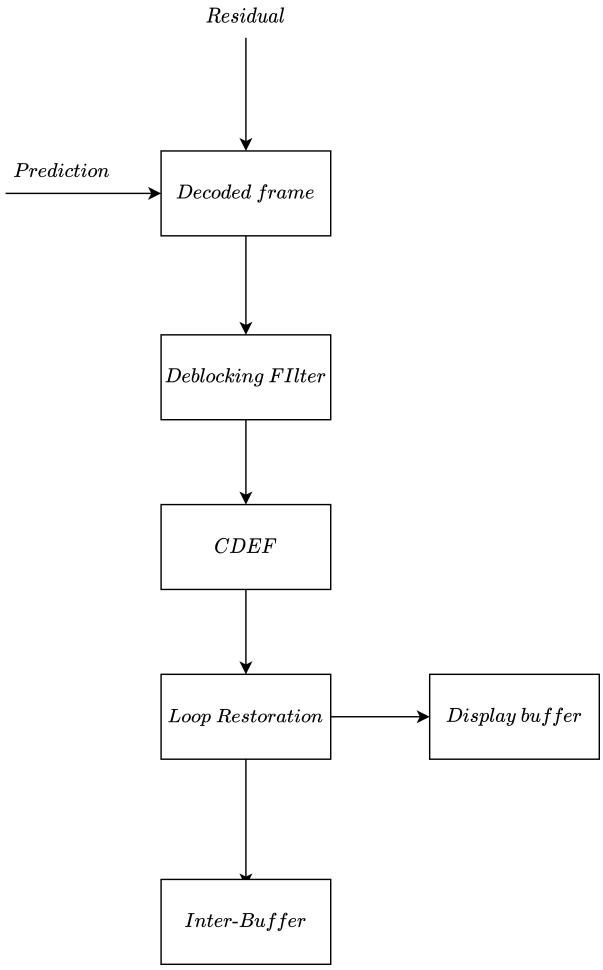

Figure 2-4: Post-processing filters.

process (transform and quantization) and recovers the residual, which is finally added to the prediction to form the decoded block. However, the process mentioned above is not always lossless (QP=0). Thus, boundary artifacts between transform blocks may appear (figure **2-5**). Also, the deblocking filter tool is implemented to mitigate boundary artifacts.

The deblocking algorithms use vertical and horizontal FIR low-pass filters with 4, 8, or 14 taps for the luma component and 4 or 6 for chroma. The size of the FIR filters relies on the minimum transform size between the blocks sharing the boundary. For example, figure **2-6** shows a case where the dimension of B-block determines the size of the filter. To avoid wrongly blurring natural edges, the deblocking tools implement a series of thresholds to determine if the filter is applied or not. The conditions are: 1) $\mid p_1 - p_0 \mid > T_0$; 2) $\mid q_1 - q_0 \mid > T_0$; 3) $2* \mid p_0 - q_0 \mid + \frac{\mid p_1 - q_1 \mid}{2} > T_1$; 4) $\mid p3_1 - p_2 \mid > T_0$, and 5) $\mid q3_1 - q_2 \mid > T_0$. The last two (4,5)

Figure 2-5: Boundaries artifacts cause by quantization.

are used only for filter taps 8 and 14. Figure **2-7** illustrates the position of the pixels q_x and p_x.

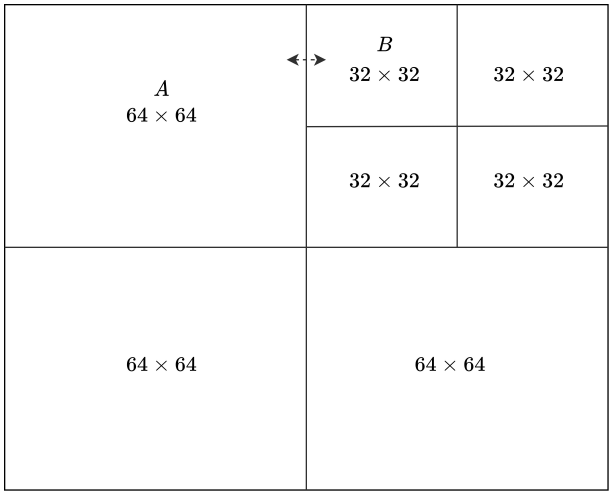

Figure 2-6: Boundary blocks to determine the size of the deblocking filter.

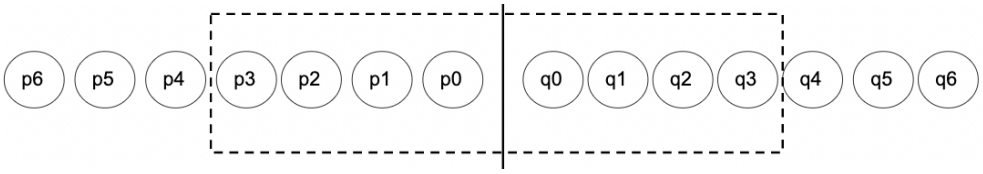

Figure 2-7: Boundary pixels involved in the deblocking filter.

2.2.2 Constrained Directional Enhancement Filter (CDEF)

CDEF [44] is designed to remove ringing artifacts (depicted in figure **2-8**) around hard edges. It is achieved by applying two filters (45° off) to each pixel. The selection of the proper

filter is performed based on the minimization of the equation (2-1).

$$E_d^2 = \sum_k \sum_{p \in P_{d,k}} (x_p - \mu_{d,k})^2, \tag{2-1}$$

Where P are the pixels in the selected direction, as shown in figure **2-9**, and μ is the mean of the group P.

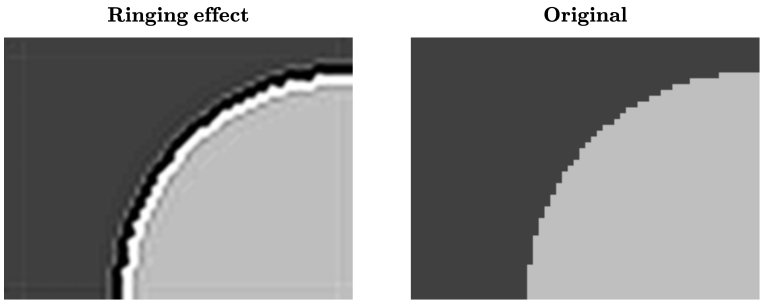

Figure 2-8: Ringing artifact [29].

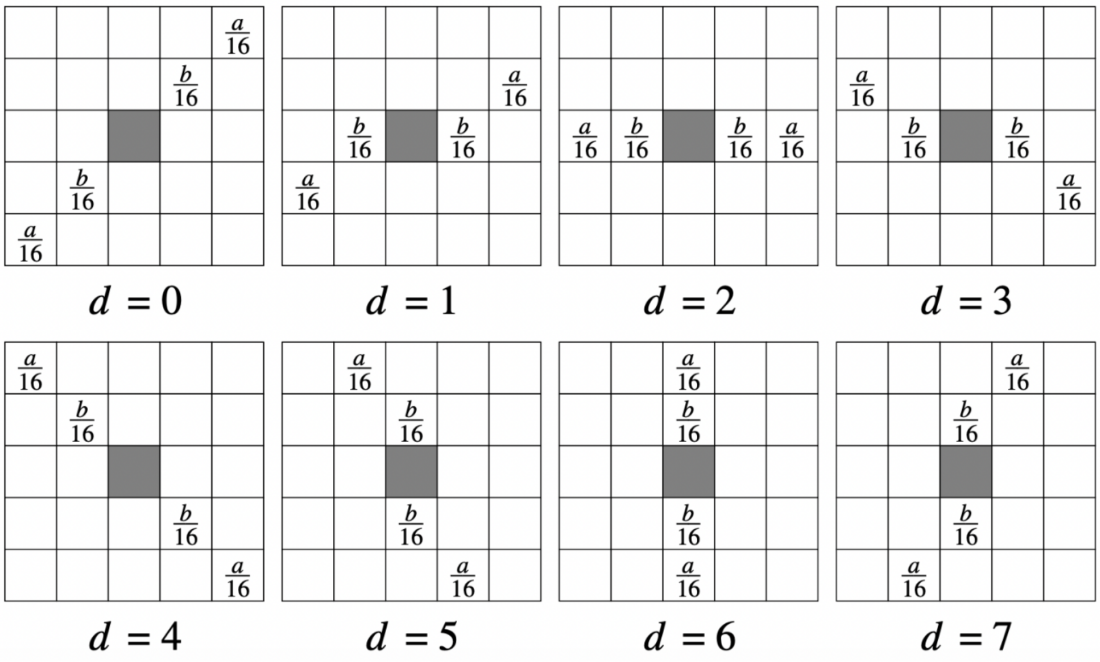

Figure 2-9: CDEF filters in 8 directions.

2.2.3 In-loop restoration filters

In-loop restoration filters are applied to loop restoration units (LRU) that can be 64×64, 128×128, or 256×256 pixel blocks. Each LRU can independently select one of three possible restoration options: 1) Wiener filter, 2) Self-guided filter [50], 3) Bypass filtering.

Self-Guided Filter

Self-Guided Filter relies on calculating two possible restored versions (X_1 and X_2) of the decoded block X. Then, it projects the mismatch between each version and the reference to obtain a unified restored patch (X_r), described in equation (2-2).

$$X_r = X + \alpha(X_1 - X) + \beta(X_2 - X) \tag{2-2}$$

Before that, X is denoised using the equation (2-3) at the pixel basis with two pairs of parameters sent from the encoder $(r_1, e_1), (r_2, e_2)$. Finally, X_1 and X_2 are computed using α and β. The self-guided filter requires four parameters to be exchanged from the encoder:

$$\hat{x} = \frac{\sigma^2}{\sigma^2 + e} x + \frac{e}{\sigma^2 + e} \mu \tag{2-3}$$

Where $\hat{x}$ is the denoised pixel, μ is the mean of the block determined by r, and e is the standard deviation of the same block.

Wiener Filter

Wiener theory [47] is a well-known restoration technique widely applied to 1-D time series systems since 1949. It uses the concept of the minimum mean square error (MMSE) to predict a signal $s(t)$, after being corrupted by a noise $w(t)$. Both signals are considered wide-sense stationary processes. Wiener initially presented two versions: causal and non-causal filters. The last case was not physically realizable for time series because it considers past, present, and future samples.

Wiener's theory started getting relevant but had not been applied to 2-D scenarios (image filtering, prediction, and smoothing). It was only in 1982 when Ekstrom [18] presented a physically realizable 2-D version of the original Wiener filter and demonstrated that it could also be extended to multi-dimensions. Ekstrom formulated the optimal error calculation to find the best parameters that define the filter configuration as described in equation (2-4):

$$H(z) = \frac{Syx(z)}{S_{xx}(z)}, \tag{2-4}$$

Where S_{xy} is the spectral energy of the correlation between X (reference) and Y (distorted) images. Similarly, S_{xx} is the spectral energy of the autocorrelation of X. The feasibility of applying Wiener on images gets relevant in video restoration problems, because it aims to find the kernel that linearly relates the original images and its distorted version. In others words, it allows the encoder to determine the kernel coefficients that the decoder can use to restore each block. This approach leads the implementation of Wiener in both AV1 and HEVC, and it is still part of the AV2. The first version implemented in AV1, presented outstanding compression efficiency, but complexity and signal information were desirable for improvement. Therefore, considering symmetricity, Siekmann *et al.* [43] proposed a separable filter that achieves a reduction of 33.33% and 50% in sum and multiplication operations for a 3×3 block. In addition, it was observed that for 9×9 block size, the efficiency is 77.77% and 80% for sum and multiplications, respectively.

Besides the operation performance, the signaling information is also drastically reduced. For example, a 9×9 block using a non-separable filter requires 81 coefficients. For a non-separable symmetric, it requires 41 coefficients; and for the case of separable-symmetric filters, it requires only 18 coefficients. In other words, it represents 78% less signaling information to send to the encoder. It was also noted that the non-separable and separable filter maintains a similar performance in terms of bit saving. The separable version is, in fact, the official implementation of the Wiener filter in the AV1 reference code. However, the processing time and complexity are still factors to improve in future approaches. Table **2-2** presents performance results of Wiener Filter vs Passthrough mode using AV1, over ten frames (1920×1080).

	ON	OFF	Performance ON vs OFF
Average speed (fps)	22	15	−32%
Total encoding time (ms)	27951	40856	+46%

Table 2-2: Wiener Filter vs Passthrough mode in AV1.

2.3 Deep-learning restoration

In-loop restoration could be classified as part of the non-blind image processing task. However, in the context of video compression this problem is even more particular. The estimation of the blurring kernel is not part of the essential objectives, since the distorted and reference frames are always known on the encoder side. In this case, the approaches to tackle this problem from the perspective of deep-learning architectures become more limited and point to video coding applications in most cases. At first, Jia *et al.* [25] proposed a content-aware CNN-based (CAC) strategy compound by a block-based model selection and restoration modules incorporated in the HEVC reference code. The first component implements a discriminative network to select the most appropriate CNN [35] per each CTU (Coding tree Unit). The network structure is depicted in figure **2-10**.

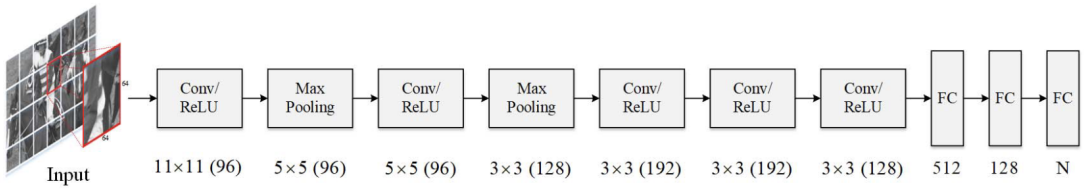

Figure 2-10: CNN discriminative net architecture [25].

The selection of which CNN is used, relies on the previous labeling of the training content, where several categories are considered. Therefore, the discriminative network chooses the CNN, which minimizes the loss function described in the equation (2-6), considering the specific content:

$$J_{CTU} = \Delta D_{CTU} + \lambda R, \tag{2-5}$$

$$\Delta D_{CTU} = D'_{CTU} - D_{CTU}, \tag{2-6}$$

Where J_{CTU} refers to the performance of the n-sima CNN, Δ_{CTU} is the variation in terms of BD-rate between the CTU before and after the restoring process, λ is the lagrange multiplier and controls the tradeoff between rate and distortion. Finally, R represents the required amount of bits for signaling. The method (CAC) has been evaluated against the reference codec HEVC, obtaining performance improvements in BD-rate, between 2 % and 4 %. Complexity is also considered against VDSR [27], VRCNN [7], and ALF [6]. The results demonstrate that the current non-DL-based algorithm (ALF) embedded into HEVC

is still the most efficient, achieving 123 % vs. 11656 % (CAC) in terms of decoder complexity.

Considering the previous results, it is straightforward to identify that quality can be enhanced. However, complexity is still far from reaching the expected level of the future video codec standards (<10x). Inspired by this fact, Ding *et al.* [9] proposed a CNN-based method (SimNet) that reduces the complexity by leveraging a skipping strategy where a simple CNN network restores specific frames, and the remaining frames continue using the traditional restoration non-DL-based method. This proposal is being formally evaluated to be part of the next AV2 video codec. SimNet applies restoration on Intra and Inter frames. The consideration of Inter frames relies on the propagation impact those have in the group of pictures (GOP). SimNet contains N cascading convolutional layers and a ReLu at last (figure **2-11**). The depth of the networks depends on the QP (Quantization Parameter) of each block.

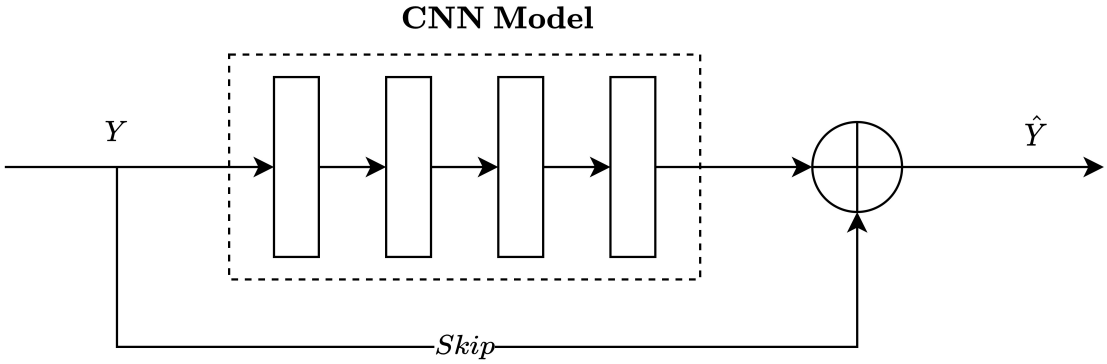

Figure 2-11: CNN in-loop filter architecture [25].

SimNet reported overall performance improvements in BD-rate, of 7.27 % and 5.47 % for Intra and Inter frames, respectively, against the reference codec AV1. It also demonstrates a processing time reduction of 12.65 % compared to AV1. However, the original paper does not provide a report of decoder complexity. The authors mentioned that they expect around 30x decoder added complexity, during the standardization meetings.

The original publication of SimNet highlights an over-filtering problem closely related to the effect caused by Inter frames that are filtered and then used as a reference for next frames. The effect is a propagated noise that reduces the performance of the CNN-based restoration methods in comparison to AV1. SimNet tackles this issue by skipping the frames that potentially introduce errors. Considering this problem, D. Ding (2020) proposes a transfer-

learning method for only-Inter frames that incorporates back the reconstructed frames into the training set of the CNN model. This algorithm achieves higher visual quality against the reference code HEVC, and increases the processing time by around 4%. The encoder complexity is not reported, but we infer that it may consume around 5x SimNet due the progressive training.

Kong *et al.* [28] presents a Guided CNN architecture with similar outcomes, BD-rate : 1.84 % - 3.06 % and additional processing time, 23.79 % (figure 6). The restoration lies in a linear combination of the N CNN's weighted outputs. Each CNN aims to capture distinct features of an image. However, the paper does not specify which characteristics capture from an image. The idea behind is to have a lightweight array of N CNN and obtain the optimal vector of weights that minimize the loss function described in equation (2-7).

$$r_{corr} = a_0 r_0 + a_1 r_1 + ... + a_{M-1} r_{M-1} \tag{2-7}$$

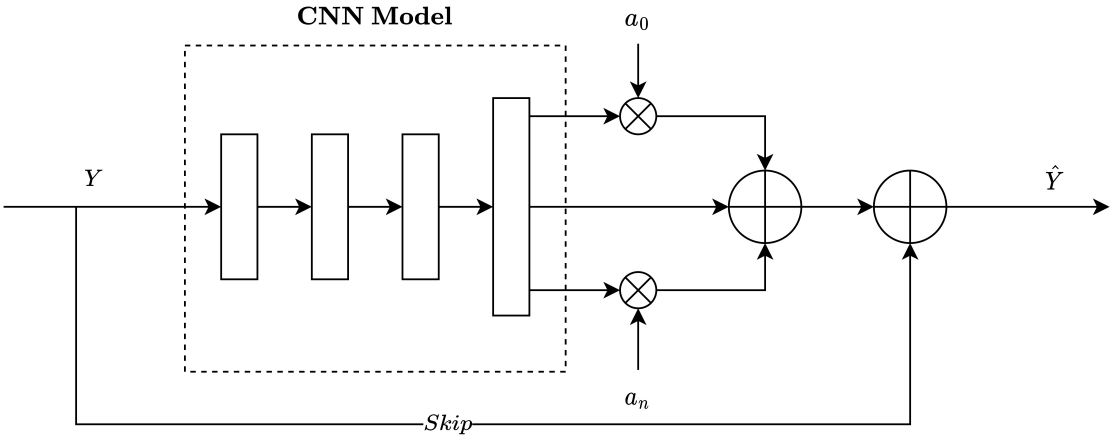

Figure 2-12: Guided CNN Restoration (GNR) [28].

3 Sparse representation

Let be X a $\sqrt{n} \times \sqrt{n}$ gray-scale image represented as a linear combination of m weighted columns vectors $v_i \in \Phi$, with dimension in $n \times m$, as described in equation (3-1). If Φ is properly defined [16], a vector α with few nonzero coefficients representing the weighted values can be obtained. This idea is the basis of sparse representation [15] and has been successfully exploited on different image processing problems, such as: restoration [31, 12], deblurring [14, 13], denoising [19, 24, 40] and super-resolution [17, 26, 49].

$$X \cong \alpha_1 v_1 + \alpha_2 v_2 \; .. \; + \alpha_m v_m. \tag{3-1}$$

Contrasting classical methods such as Inverse Fast Fourier Transform (IFFT), the dictionary Φ is not necessarily invertible. A essential characteristics of sparse representation lies in the high redundancy of Φ ($m >> n$), which enables the sparsity of a vector α [39]. However, high redundancy of Φ brings an infinity of possible solutions, which opens a question about the most accurate solution (if there is one), and even more when noise is presented. This question has been addressed for awhile and is translated into a more formal approach as an optimization problem described in equation (3-2):

$$\alpha = ||X - \alpha\Phi||_2^2 + \lambda||\alpha||_0, \tag{3-2}$$

Where X represents the reference image, and λ is a regularization factor. Using a formal approach, we aim to find a vector α and a dictionary Φ that minimizes the distance of $\alpha\Phi$ to X while maintaining a minimum number of nonzero coefficients in α. The original definition utilizes a Euclidean distance or L_2-*norm* for modeling the error and L_0-*norm* to penalize the number of nonzeros coefficients in α. Donoho *et al.* [15] showed that the L_1-*norm* could be used instead of L_0. Converting the NP-hard combinatorial problem into a computational realizable problem. Therefore, equation (3-2) becomes in equation (3-3):

$$\alpha = ||Y - \alpha\Phi||_2^2 + \lambda||\alpha||_1. \tag{3-3}$$

Considering equation (3-3), we have a realizable problem and aim to find a sparse vector α and a dictionary Φ simultaneously. Typically, sparse vectors are calculated on block-images basis, and a dictionary is pre-trained or pre-defined according to a specific application.

3.1 Basic formulation of sparse coding

Despite being an NP-hard problem, the L_0 regularization is still a suitable solution for scenarios with previous information or precondition establishing very few nonzero coefficients. In that case, the suite of greedy algorithms is commonly used and especially the Orthogonal Matching Pursuit (OMP), which was introduced by Cai *et al.* [5] and it is described in table **3-1**.

The equation (3-2) is equivalent to equation (3-4):

$$\min_{\alpha} ||\alpha||_0 \quad s.t. \quad Y = \Phi\alpha \tag{3-4}$$

In contrast, when the sparse vector size increases, and the objective is to obtain the sparseness solution, the L_1 norm performs better. The regularization problem is established following the Least Absolute Shrinkage and Selection Operator (LASSO [23]) and described in equation (3-5). Furthermore, several optimization techniques have been proposed for the solution of LASSO, such as LARS, Coordinate Descent, and Feature-sign search algorithm, among others. Table **3-2** describes LARS.

$$\min_{\alpha} ||Y - \alpha||_2 + \lambda||\alpha||_1. \tag{3-5}$$

A variety of optimization methods have been proposed for the solution, such as LARS, Coordinate Descent and Feature-sign search algorithm, among others.

3.2 Dictionary

We found in the literature vast of articles about dictionary definitions and learning. After a deep search, we select three methods that are align with the scope of our approach; which have three main pre-conditions: 1) Dictionaries or associated signaling information must be as small as possible or unnecessary because it impacts the bit budget required for decoding. 2) The dictionary learning process has to rely on the encoder side. 3) The processes on the decoder side, regarding dictionary selection and update, have to keep low complexity.

Input: The vector $Y \in R^{\sqrt{n}}$, the matrix $\Phi \in R^{\sqrt{n}\times m}$, and the termination threshold for the residual norm e. **Output:** The sparse vector $\alpha \in R^m$. **Task:** Approximate the vector y by using the fewest columns of the matrix Φ.
1. Initialization: $\alpha_0 = 0$, $r_t = y$, $t = 1$, $I = \{\}$ 2. Compute the correlation vector: $c_t = \Phi^T r_{t-1}$ 3. Find a column index of the matrix Φ that is best correlated with current residual vector. This can be achieved by determining the index of the largest absolute entry in the vector. $i = \underset{j\in I^c}{\operatorname{argmax}} \mid c_t(j) \mid$ where I^c is the inactive set (the set have indices of columns of the matrix Φ that are not in the active set). 4. Add i to the active set: $I = I \cup \{\}$ 5. Solve the least square problem: $\Phi_I^T \Phi_I \alpha_t(I) = \Phi_I^T y$ 6. Compute the new residual vector : $r_t = y - \Phi\alpha_t$ 7. If $\|r_t\|_2 < e$, terminate and return $\alpha = \alpha_t$ as the final solution. Else, increase the iteration. Counter: $t = t + 1$ and return to step 2.

Table 3-1: OMP Algorithm [10]

3.2.1 Concatenation of two orthogonal basis

The groups actively working on standardization, often consider models the residual between a reference and distorted frames in some sparse transform domain, such as the Wavelet or Fourier [1]. However, there is a solid experimental understanding that, in several cases, images combine elements from different transforms [16]. As a manner of illustration, let X be an image represented by a sparse vector α and an orthogonal matrix Φ. A second representation of X using a sparse vector β and a matrix Ψ. Both matrices are $\sqrt{n} \times \sqrt{n}$, as described in equation (3-6):

$$X = \alpha\Phi = \beta\Psi. \tag{3-6}$$

If Φ and Ψ are concatenated to form a single $\sqrt{n} \times 2\sqrt{n}$ dictionary, instead of $\sqrt{n} \times \sqrt{n}$

Input: The vector $Y \in R^{\sqrt{n}}$, the matrix $\Phi \in R^{\sqrt{n} \times m}$, and the termination threshold for the residual norm e. **Output:** The sparse vector $\alpha \in R^m$. **Task:** Approximate the vector y by using the fewest columns of the matrix Φ.
1. Initialization: $\alpha_0 = 0$, $r_t = y$,$t = 1$,$I = \{\}$ 2. Compute the correlation vector: $c_t = \Phi^T r_{t-1}$ 3. Compute the maximum absolute value in the correlation vector: $\lambda_t = \|c_t\|_\infty$ 4. If λ_t is zero or approaches a very small value, LARS is terminated and the vector α_t is returned as the final solution, otherwise the following steps are implemented. 5. Find the active set: $I = \{j :\mid c_t(j) \mid = \lambda_t\}$ 6. Solve the following least square problem to find active entries of the updated direction: $\Phi_I^T \Phi_I d_t(I) = sign(c_t(I))$ where $sign(c_t(I))$ returns the sign of the active entries of the correlation vector c_t. 7. Set the inactive entries of the updated direction to zero: $d_t(I^c) = 0$ 8. Calculate the step size: $\gamma_t = \min_{i \in I^c} \{\frac{\lambda_t - c_t(i)}{1 - a_i^T v_t}, \frac{\lambda_t + c_t(i)}{1 + a_i^T v_t}\}$ Where $v_t = \Phi_I d_t(I)$ 9. Update the solution vector: $\alpha_t = \alpha_{t-1} + \gamma_t d_t$ 10. Compute the new residual vector: $r_t = y - \Phi \alpha_t$ 11. If $\|r_t\|_2 < e$, terminate and return $\alpha = \alpha_t$ as the final solution. Else, increase the iteration counter: $t = t + 1$ and return to step 2.

Table 3-2: LARS Algorithm

matrices[1], we obtain a new representation: $X = [\Phi\ \Psi]\gamma$. Unlike equation (3-6), where $\alpha = X\Phi^T$ and $\beta = X\Psi^T$, the representation of $X = [\Phi\ \Psi]\gamma$ using an overcomplete dictionary leads to infinite solutions. However, Elad *et al.* [20] demonstrate a condition to determine the upper limit of the number of nonzeros coefficients in γ without requiring to execute any complex processing. Furthermore, Elad *et al.* state that under the condition depicted in equation (3-7), the solution of the optimization problems described in equation (3-2) and (3-3), that constrain the sparsity throughout the L_0 and l_1 norms respectively, converge to the same result. It is a remarkable statement considering the computational unfeasibility of

[1]Donoho *et al.* defines the concept of a dictionary as a combination of N-by-N basis

the l_0-norm.

$$\|\gamma\|_0 < \frac{0.9142}{M}, \tag{3-7}$$

Where M is the maximum absolute value of the cross inner product between the basis Φ and Ψ.

block size	$M = Sup \mid< \phi_i, \gamma_j >\mid \vee(i,j)$	$\|\alpha_i\|_0 < \frac{0.9142}{M}$
4×4	0.9061	1.0089
8×8	0.9007	1.0150
16×16	0.9003	1.0154
32×32	0.9003	0.9003
64×64	0.9003	1.0154

Table 3-3: Optimal total sparse non-zero coefficients using DCT+DWT dictionary.

Table **3-3** presents the optimal total sparse non-zero coefficients using DCT+DWT dictionary using different block size.

3.2.2 Multi-dictionaries and dynamic selection

Dong *et al.* [11] state that a single $\sqrt{n} \times \sqrt{n}$ matrix can effectively represent an image-patch X. Nonetheless, instead of using a universal basis, such as Wavelet or Fourier, Dong propose a set of dictionaries with specific high-frequency characteristics and groups them by the K-means cluster algorithm. Before executing any procedure, the first step consists of building a dataset with high-quality image patches from the YUV space that are neither compressed nor distorted. The idea is to collect several features at the block level to create a rich and heterogeneous universe of patches that can eventually recover any image patch. In the second step, they apply a high-pass filter to all patches for obtaining a high-frequency version of each. Since, dictionaries perform well on recovering image edges and high color variations that are the most relevant for the human visual system. However, the proposed methodology includes a dictionary built from the low-passed filtering patches clustering to cover the plain and low-variation features. Once clusters are ready, the next step is to obtain a compact version of dictionaries per cluster. Dong found that if the algorithm efficiently selects a proper group, the dictionaries can be $\sqrt{n} \times \sqrt{n}$ (non-overcomplete), reducing the optimization problem to a simple matrix-vector multiplication. Consequently, the algorithm

utilizes the classical signal decorrelation, and dimensional reduction technique PCA [30] to obtain a set of K dictionaries with dimensions $\sqrt{n} \times \sqrt{n}$ as described in equation (3-8):

$$\Phi = \{\Phi_0, \Phi_1, \Phi_2, ..., \Phi_k\} \tag{3-8}$$

The methodology contains a high-processing stage during the dictionary learning process, but it is executed once and offline. Conversely, the proper dictionary selection is performed per image patch basis and dynamically.

3.2.3 Universal dictionary

Yang *et al.* [49] presents a dictionary learning technique for image super-resolution, where they train both low Φ_l and high Φ_h resolution dictionaries simultaneously. This algorithm is an extension of the classical single dictionary learning method described in equation (3-9):

$$\min_{\{\Phi_l, \Phi_h, Z\}} \|X_c - \Phi_c Z\|_2^2 + (\frac{1}{N} + \frac{1}{M})\|Z\|_1 \tag{3-9}$$

In equation (3-9), Φ_c integrates information about Φ_l and Φ_h. In the same manner, X_c concatenates the HR (X_h) and LR (Y_l) patches in equation (3-10):

$$X_c = \begin{bmatrix} \frac{1}{N} X_h \\ \frac{1}{M} Y_l \end{bmatrix}, \ \Phi_c = \begin{bmatrix} \frac{1}{N} \Phi_h \\ \frac{1}{M} \Phi_l \end{bmatrix} \tag{3-10}$$

Where N and M are the dimensions of the high-resolution and low-resolution image patches, in vector form, using a dataset of 100,000 LR and HR image patches. Yang *et al.* state a more robust relation between LR and HR based on a feature extraction process of the LR. They apply four filters to extract different characteristics and concatenate them to construct the LR image blocks for training and validation. In this way, the LR incorporates information about neighbors, proving more than a simple linear relation with the HR block. Conversely, they obtain the HR image patches using the bicubic interpolation method with a factor of 2x.

4 Sparse In-loop Video Coding Restoration Method (SRM)

The problem of in-loop restoration is presented in chapter 2, including the current methods, available in the reference AV1 video codec, and the most relevant proposals using deep learning, that are potential tools for AV2. Regarding existing methods, there is an extensive effort to optimize the performance at the complexity and BD rate levels, *i.e.* Wiener and a well-stated theoretical non-blind filter, it is separated into vertical and horizontal components, that together with the concept of symmetry – introduced by Siekmann *et al.* [43] – give a powerful restoration mechanism at the expenses of few side bytes. For instance, a 256×256 patch only requires three taps coefficients that can be represented with 12 bytes (96 bits) using double precision. The case of the self-guided is similar, where only four coefficients are necessary. Despite the benefits of both approaches (approx. 1.338 BD rate gain), there is still room for further optimization –mainly in processing time.

On the other hand, leaning-based algorithms such as content-aware CNN proposed by Jia *et al.* [25] achieve a nearly 4 % gain in terms of BD rate with a 113 % (encoder) and 11656 % (decoder) complexity compared to anchor HEVC. Another case is the self-guided CNN method presented by Ding *et al.* [9] that reports an attractive BD-rate gain of around 1 %-3 %, that does not require side information. However, the complexity of the residual prediction in the encoder (+23 %) is still higher than in the existing methods. Ding *et al.* do not report decoding complexity, but considering the prediction network, it is expected to be similar to encoding. Finally, we summarize in table **4-1**, the performance of the classic Wiener and the most relevant approaches based on deep learning.

Considering the opportunities in the field of in-loop restoration, we introduce a novel learned-based sparse approach supported by three main components: 1) Sparse decoding residual, 2) Sparse coefficients estimator, and 3) Sparse position estimator, which are further detailed

Characteristic	Wiener	Guided CNN	CNN[1]
Encoder complexity (frames/sec)	7.48	9.26	8.79
Decoder complexity	low	high	high
BD-rate	-1.338	-2.85	-1.39

Table 4-1: Characteristics of Traditional vs Deep Learning In-loop Restoration approaches.

in this Chapter. Figures **4-1** and **4-2** present the overall architecture of our method.

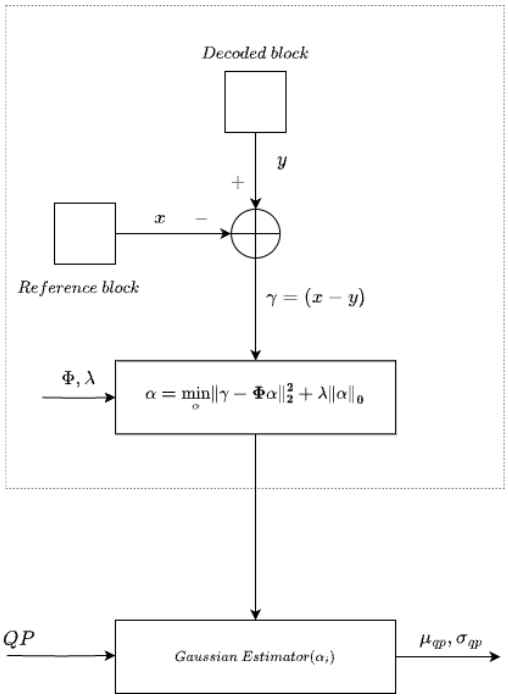

Figure 4-1: Sparse decoded - Gaussian

4.1 Sparse decoding residual

During the video compression process, the spatial residual between the reference and Inter/Intra predicted frame is transformed into the frequency domain. Then, the resulting coefficients, such as DCT, are mapped into predefined quantized factors, which vary from 0-255, where 0 is lossless. After that, an entropy operation is applied to eliminate the statistical redundancy. In the particular case of DCT, it means bringing more bits to represent the DC and the high-frequency coefficients. Finally, the bit sequence is sent to the decoder.

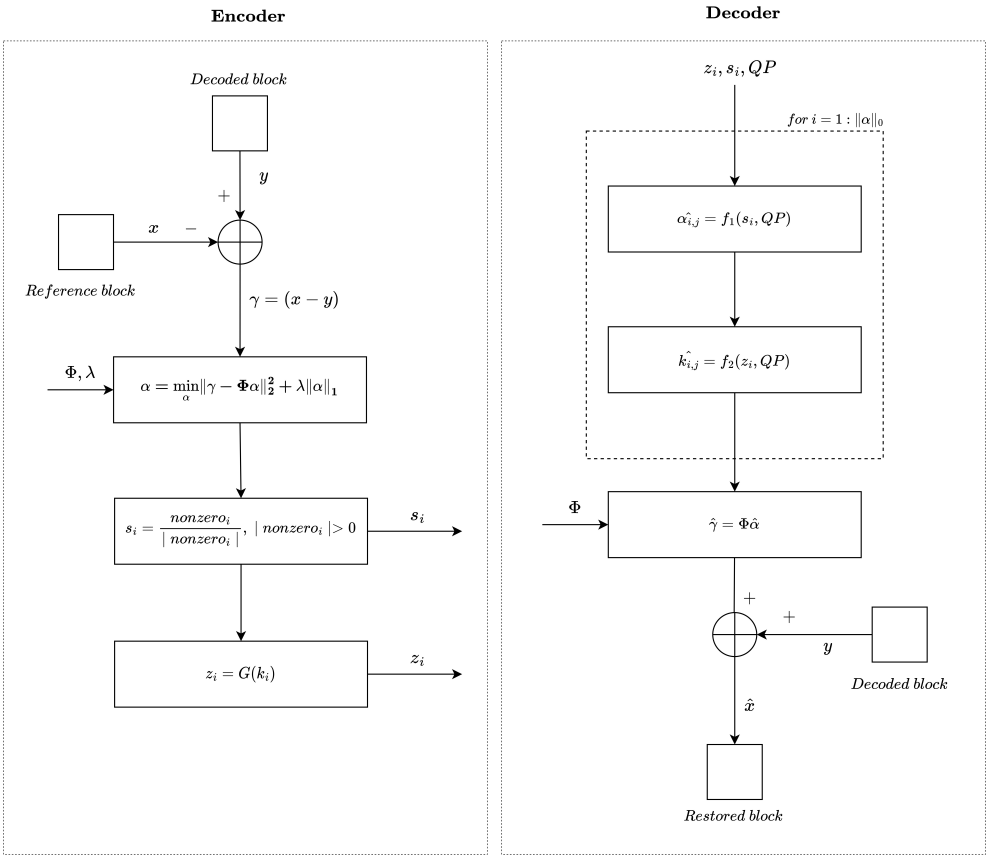

Figure 4-2: Encoder/Decoder sparse in-loop restoration

Equation 4-1 models the frequency-domain residual after quantization Q [42] and transform T operations:

$$g = Q[T\{r\}] \tag{4-1}$$

Figure **4-3** illustrates the process described above. In order to recover the original frame, the decoder uses the prediction frame and the residual. The last is obtained after applying inverse transform and quantization operators to the entropy-decoded bitstream. However, quantization is a lossy task; depending on the number of levels, such as 85, 110, or 210, the loss is higher. That is something expected by the trade-off between bitrate and quality. Therefore the result of the inverse transform is not equal to the residual r (equation 4-2):

$$\hat{r} = T^{-1}[Q^{-1}\{g\}] \tag{4-2}$$

$$r \neq \hat{r}$$

Since the quantization error (e_q) is linear, we can approximate r as follow:

$$r \approx T^{-1}[Q^{-1}\{g\} + e_q] \tag{4-3}$$
$$r \approx \hat{r} + T^{-1}[e_q]$$

As illustrated in figure **4-3**, the reference frame X is theoretically expected to be equal to its prediction X_{pre} plus a residual r. Integrating this into equation 4-4 conducts to express the relation between the decoded frame Y and its reference X in equation 4-4:

$$X \approx X_{pre} + r \tag{4-4}$$
$$X \approx X_{pre} + \hat{r} + \mathbf{T^{-1}}[\mathbf{e_q}]$$
$$X \approx Y + \mathbf{T^{-1}}[\mathbf{e_q}]$$

In the decoder, X_{pre} and $\hat{r}$ are always obtained through prediction and inverse quantization/transformation operations, respectively. Therefore, the proposed method models the differential cause by the inverse transform of the quantization error ($T^{-1}[e_q]$). From now on, we call the differential **decoding residual (DR)**. The reference AV2 video codec implements a series of tools, described in chapter 2, to either reduce the effect of the quantization on the block boundaries or recover the lost information (same as DR). Those processes require few bytes to improve the visual quality of the decoded frame, then the final bitrate is barely impacted. Our method takes into account this constraint –low impact in the final bitrate– by modeling DR using the sparse theory to rely on a few nonzero coefficients expanded by a proper basis or dictionary.

In our proposal, we use γ to represent DR and the L_0 norm presented in Equation 3-4 to obtain the equation (4-5):

$$||\alpha||_0 \leq z \quad s.t. \quad \gamma = \alpha\Phi \tag{4-5}$$

The use of the L_0 norm and the OMP algorithm relies on our target, which is to minimize the total of nonzeros instead of the error. The solution of the optimization problem in equation (4-5) provides a sparse vector α that, together with the dictionary Φ, can estimate the residual $\hat{\gamma} = \Phi\alpha$. The calculation of α is entirely executed in the encoder. The resulting few nonzero coefficients and corresponding positions in the vector are sent to the decoder as restoration signaling information. The residual γ is split into 8×8, 16×16, or 32×32 blocks, which are simultaneously processed in order to make the problem suitable in terms

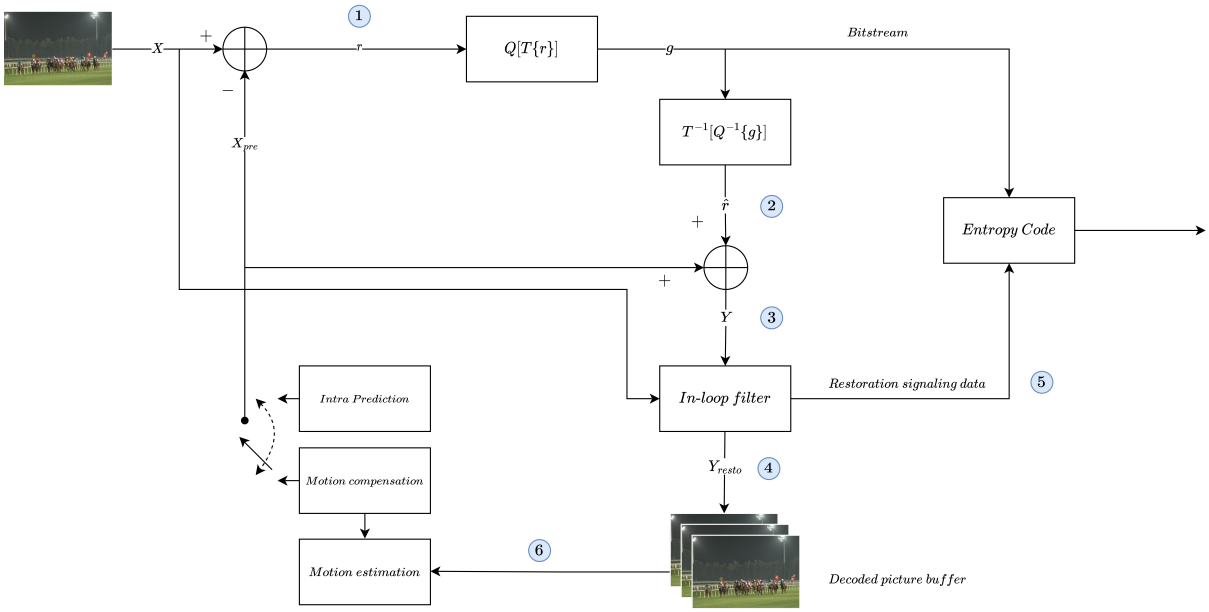

Figure 4-3: Simplified video codec architecture.

of computer memory usage. An operator $R_{i,j}$ is introduced to extract the i, j patch of size $\sqrt{n} \times \sqrt{n}$. The equation (4-6) models the i, j sparse vector α:

$$||\alpha_{i,j}||_0 \leq z, \quad s.t. \quad R_{i,j}\gamma = \alpha_{i,j}\Phi \tag{4-6}$$

The patch-based approach requires blocks to be overlapped to avoid edge reconstruction artifacts. Mairal *et al.* [32] introduce a weighted average formulation that we adapt in equation (4-7):

$$\hat{\gamma} = \left(\sum_{i,j} R_{i,j}^T R_{i,j}\right)^{-1} \left(\sum_{i,j} R_{i,j}^T \Phi \alpha_{i,j}\right) \tag{4-7}$$

We evaluated three dictionary approaches, described in section 3.2, to select the proper vector-basis to model γ:

1. The first method is concatenating two orthogonal basis [20]. We combined the discrete cosine transform (DCT), discrete wavelet transform (DWT), and identity matrix (I) to obtain five vector-basis.

2. The second method relies on a universal dictionary [49]. We used the A2 raw video sequence (Described in table **5-1**) and the respective decoded frames with $QP = [85, 160, 210]$ to train four dictionaries (1 lossless and 1 per QP). During the testing

and evaluation, we used a concatenation of the lossless and the respective QP dictionary to obtain dimensions of 64×256 and 256×1024, for blocks size of 8×8 and 16×16 respectively.

3. Finally, we use the multi-dictionaries method [11] with 10 and 20 clusters using dictionaries of dimensions 64×64 and 128×128.

Details of each basis are depicted in table **4-2**, and the performance evaluation, in terms of PSNR, is presented in table (**4-2**). Further information on the experiments is covered in chapter 5.

Dictionary	Description
Φ_1	$DCT + I$
Φ_2	DCT
Φ_3	$DCT + DWT$
Φ_4	DWT
Φ_5	$DWT + I$
Φ_6	Universal
Φ_7	Multi-dictionaries (10 clusters)
Φ_8	Multi-dictionaries(20 clusters)

Table 4-2: Evaluated Dictionaries.

	Ref	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7	Φ_8
Average	36.03	40.24	39.75	39.85	38.25	39.08	**41.23**	38.60	38.70
PSNR gain(%)		11.67	10.32	10.60	6.14	8.46	**14.41**	7.11	7.39

Table 4-3: PSNR (summary) after restoration over raw video sequence A2.

The results presented in table **4-2** and extended in table **5-7** give us several insights: 1) Universal dictionary (Φ_6) behaves better than the others with a nearly 14 % average PSNR gain across all configurations. This performance is reasonable due to the overcomplete factor of the dictionaries. 2) The multi-dictionaries methods (Φ_7,Φ_8) show that the number of clusters impacts the quality gain. A more significant number of clusters (> 20) may bring better results. However, the added computational complexity during the dictionary selections

makes this approach unsuitable for our application when the number of clusters increases. 3) All the concatenated dictionaries (Φ_1, Φ_2, Φ_3) that include DCT show gains between 10-11 %. That aligns with the previous knowledge of the most common transform basis used by AV2 and the modeling of DR previously described, in this chapter. 4) Despite the performance of the sparse representation approach, the number of bits required to signal the nonzero coefficients in the sparse vector makes it an unfeasible strategy from the perspective of BD-Rate, *i.e* using blocks of 16×16, 2 nonzero coefficients and a dictionary 256×512 requires approx 2048 bytes for restoring a 256×256 image block. Which is 170x more expensive than the current Wiener filter.

Considering the insights, we introduce a strategy to reduce the signaling information by predicting the nonzero coefficients and their position in the sparse vector. Consequently, our strategy selects the DCT basis as the dictionary based on the solid statistical redundancy regarding the magnitude of nonzero coefficients and their location in the transformed vector (sparse vector), as depicted in figure **4-6**.

4.2 Sparse coefficients estimator

In order to exploit the statistical redundancy of the sparse coefficients, we decided to use the DCT basis as dictionary in our method. In this case, the sparse vector is interpreted as the truncated version of $T[\gamma]$, where T is the DCT transform operator. Therefore, the nonzero coefficients follows predictable statistical behavior [37]. In fact, AV1/AV2 models the magnitude of the DCT coefficients as a Laplace distribution [36], which permits estimating the rate-distortion per different transform configurations. Thus, our method follows a similar assumption, but finding that a Gaussian distribution (equation 4-8) best fits the absolute magnitude of the DC nonzero coefficient (z) and the Gamma [48] distribution (equation 4-9) for the rest of the coefficients. The only variation concerns $QP = 85$ where the Laplace distribution(equation 4-10) better fits the AC coefficient's absolute value:

$$f(\mid z \mid) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{|z|-\mu}{\sigma}\right)^2} \tag{4-8}$$

$$f(z \mid a, b) = \frac{z^{a-1} e^{-bz} b^a}{\Gamma(a)} \tag{4-9}$$

$$f(z \mid \mu, b) = \frac{1}{2b} e^{\left(-\frac{|z-\mu|}{b}\right)} \tag{4-10}$$

The parameters of the mentioned distributions are correlated with the QP level. Accordingly, we estimate the parameters per each $QP \in \{85, 110, 135, 160, 185, 210\}$, as shown in table **4-4**.

QP	DC	AC
85	$--$	$Laplace(\mu = 6.01, b = 1.08)$
110	$\mathcal{N}(a = 13.25, b = 0.70)$	$\mathcal{N}(\mu = 10.06, \sigma = 2.77)$
135	$\Gamma(a = 6.30, b = 2.42)$	$\mathcal{N}(\mu = 17.98, \sigma = 6.94)$
160	$\Gamma(a = 4.09, b = 5.89)$	$\mathcal{N}(\mu = 33.45, \sigma = 14.68)$
185	$\Gamma(a = 3.04, b = 13.11)$	$\mathcal{N}(\mu = 61.37, \sigma = 28.53)$
210	$\Gamma(a = 2.50, b = 27.81)$	$\mathcal{N}(\mu = 110.20, \sigma = 54.29)$

Table 4-4: PDF's parameters by QP Level.

Our method estimates the nonzero coefficients following the Probability Distribution Functions (PDF) instead of sharing them with the decoder. Our method uses the sign of the coefficient after verifying if it is correct. The magnitude of the coefficient could vary but still add gain to the distorted frame –since the method is concentrated in a residual, not in a frame– as illustrated in figure **4-4**. For the case of ($QP = 135$), the cost of predicting the coefficients is -0.16 dB in terms of PSNR. However, the saving is around 25K Bytes (2.78 %) per frame of 1280×720[2].

Figure 4-4: Sparse coefficient estimation for $QP = 135$ (Distorted Image PSNR = 38.81dB).

Figures **4-5** and **4-6** present the PDF for nonzero coefficients associated with $QP = 110$ and $QP = 185$. As we can empirically identify and confirm in table **4-4**, the standard deviation of

[2]This example uses blocks of 16×16 and $\|\alpha\|_0 = 2$

$QP = 185$ is approx. $10\times$ the standard deviation of $QP = 110$. It is completely aligned with the previous knowledge that higher QPs (lower quality) require more comprehensive ranges or frequencies to compensate for the quantization loss. As a result, $QP \in [185, 210]$ introduce blocking effects during the sparse restoration process because, in some cases, the nonzero coefficient estimator overcompensates higher frequencies and consequently strengthens the block boundaries. The blocking effect is depicted in figure **5-13**. We experimentally found that reducing the prediction by a factor of 0.6 mitigates the overcompensation at the cost of about -1dB (PSNR) compared with the standard sparse restoration method when using a block size of 16×16 and two atoms per block.

4.3 Sparse position estimator

From the frequency-domain perspective, the positions of nonzero coefficients, in α, are the most relevant frequencies in the decoded block (y) requiring compensation. On the decoder side, this information is unknown. However, considering the quantization errors equally affect all the AC components of y, we can expect that the most relevant frequencies in y are also the ones requiring more considerable restoration. We confirm this idea executing restoration over patches across raw video sequences with different QP levels and finding that 80 % of the time, at least one of the three most relevant coefficients in y corresponds to the most relevant frequencies in the residual decoding block. Thus, implementing a DCT transform of y could bring this information to the decoder. However, the sign of the DCT coefficients does not keep relation with the potential sign of the nonzero coefficient on a specific position. Therefore, we introduced a novel algorithm that relies on the image quality blind assessment in the frequency domain, introduced by Saad M. *et al.* [38], to evaluate the most efficient combination, in terms of the sign of two nonzero coefficients in the sparse vector whose positions are determined by the location of the most relevant components of the DCT transform of y.

We follow the well-established definition of Natural Scene Statistics (NSS [33]), where natural images are characterized by a Generalized Gaussian Distribution (GGD), and the effect of distortions, such as JPEG blocking and blur, affects the shape and histogram of the original image. However, we are not willing to evaluate the image quality. Instead, we use the features of the GGD as criteria to determine the proper predicted decoding residual block. Specifically, we utilize the GGD constant parameter a described in the equation 4-11. This

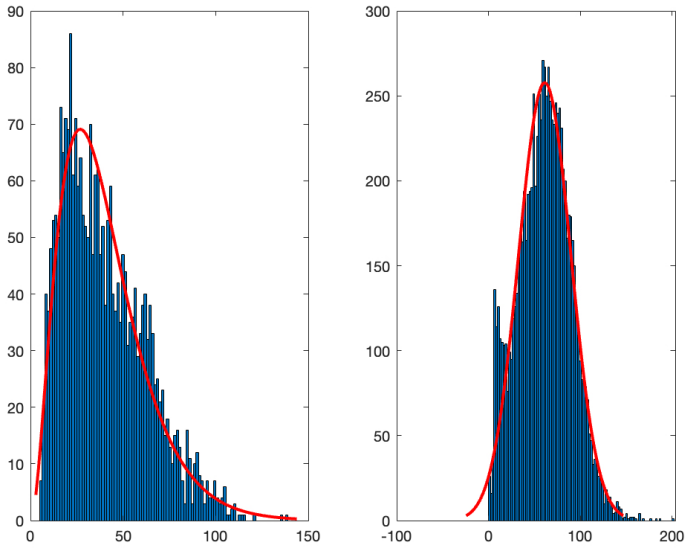

Figure 4-5: PDF for the sparse nonzero coefficients ($QP = 185$).

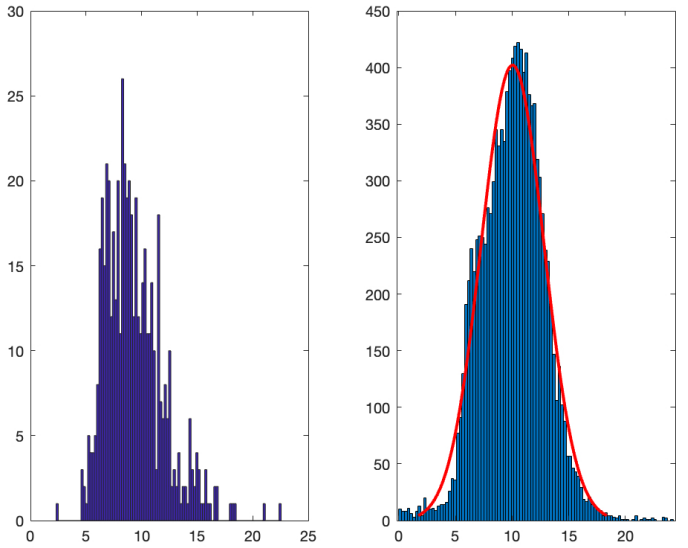

Figure 4-6: PDF for the sparse nonzero coefficients ($QP = 110$).

allows us to avoid signaling information between the encoder and decoder to share coefficients information, besides a single bit per restoration block to guide the restoration process. The complete details of our prediction algorithm are given below in tables **4-5** and **4-6**.

$$F_X(x, \mu, \sigma^2, \tau) = ae^{(-b|x-\mu|)^\tau} \quad x \in \Re \tag{4-11}$$

$$b = \left(\frac{1}{\sigma}\sqrt{\frac{\Gamma(3/\phi)}{\Gamma(1/\phi)}}\right) \quad a = \left(\frac{b\phi}{2\Gamma(1/\phi)}\right)$$

Input: Reference block: x, Decoded block : y, block size : $(\sqrt{n} \times \sqrt{n})$ **Task**: Set $encoder\_flag$
1. Apply DCT to x and y $x^f = T[x],\ y^f = T[y]$:
2. Calculate the GGD feature a for x and y: $a_x = \left(\frac{b_x\phi}{2\Gamma(1/\phi)}\right) \quad a_y = \left(\frac{b_y\phi}{2\Gamma(1/\phi)}\right)$
3. Set $encoder\_flag$: $\frac{a_x}{a_y} > 1 \rightarrow\ encoder\_flag = 1$ $\frac{a_x}{a_y} < 1 \rightarrow\ encoder\_flag = 0$

Table 4-5: Sparse prediction algorithm at the encoder per block-basis.

The comparison of our model is presented in table **4-7** across video sequences A2-A5 and B1. We use the Bjontegaard Delta-Rate (BD-Rate) [4] to evaluate our method's performance under different bitrate and quality conditions. Despite the efficiency of prediction, regarding accuracy and reduction of signaling bits, the encoder must inform the decoder whether a block is subjected to restoration or not. Our method is partially blind, which can restore blocks accuracy in most cases (up to 70 %). For the remaining 30 %, the encoder must inform the decoder that restoration is not demanded. A guiding bit ($encoder\text{-}flag$) is also added for those patches subjected to restoration. Therefore, a frame from group A2 (1920×1080) and block size 32×32 requires approx. 379 bytes for signaling, considering 50 % of the blocks are restored. This number is 2x the demanded by the switchable filter.

Another exciting result is that our method performs better on SSIM and VMAF metrics which, according to the MSU Graphics & Media Lab Video Group [3], are 90.57 % and 93.86 % (respectively) correlated to the human subjective score, that surpasses the 87.43 % reported for PSNR. The reason for that is that our model uses GGD features to select the most predictable decoding residual. It goes in the direction of the Natural Scenes Statistic,

which assesses the structure of the images instead of measuring the distance between pixels. SSIM and VMAF use the principles (structure of images).

Figure **4-7** illustrates a result under sequence B1. In the top-right correct, we can see the patch starts getting details of the plants. As described previously in this chapter, our method aims to reduce or increase those frequencies (in the DCT domain) that are more relevant for the image and provide more gain in terms of details. We omit DC coefficients which are not considered by the GGD features.

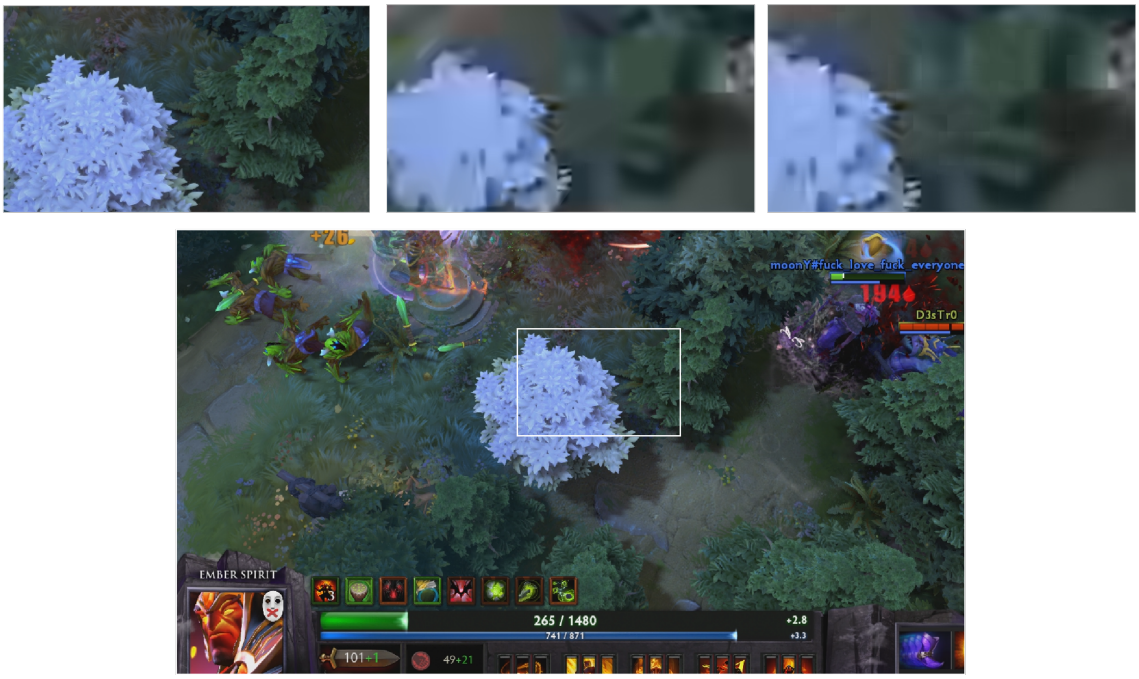

Figure 4-7: Frame restoration for sequence B1 and QP=210. Top-left : Original, Top-center: Distorted: Top-right: Distorted (+1dB gain), Bottom: Full reference image.

Input: Decoded block : y, block size : $(\sqrt{n} \times \sqrt{n})$, $encoder\_flag \in \{0, 1\}$, QP

Task: Predict the decoding residual block at the decoder (γ)

1. Apply DCT to the decoded block:

$$y^f = T[y]$$

2. Identify the position in y^f of the two coefficients with the largest absolute value

$$A = sort(\mid y^f \mid) \quad s.t. \quad \max(A) = A[0] \ \wedge \ A[0] \geq A[1]$$
$$p_0 = k_0 \quad s.t. \quad A[0] = \mid y^f_{k_0} \mid$$
$$p_1 = k_1 \quad s.t. \quad A[1] = \mid y^f_{k_1} \mid$$

3. Predict the magnitude of the sparse nonzero coefficients:

$$\{c_0, c_1\} \sim Laplace(\mu, b) \quad \forall \ QP \in \{85\}$$
$$\{c_0, c_1\} \sim \mathcal{N}(\mu, \sigma^2) \quad \forall \ QP \in \{110, 135, 160, 185, 210\}$$

4. Create four sparse vectors with the possible signs combinations

$$\forall \ i \notin \{p_0, p_1\} \ \nu_0[i] = 0, \ \nu_0[p_0] = +c_0, \ \nu_0[p_1] = +c_1$$
$$\forall \ i \notin \{p_0, p_1\} \ \nu_1[i] = 0, \ \nu_1[p_0] = -c_0, \ \nu_1[p_1] = +c_1$$
$$\forall \ i \notin \{p_0, p_1\} \ \nu_2[i] = 0, \ \nu_2[p_0] = +c_0, \ \nu_2[p_1] = -c_1$$
$$\forall \ i \notin \{p_0, p_1\} \ \nu_3[i] = 0, \ \nu_3[p_0] = -c_0, \ \nu_3[p_1] = +c_1$$

5. Obtain four potential restored blocks in the DCT domain

$$y^f_0 = y^f + \nu_0$$
$$y^f_1 = y^f + \nu_1$$
$$y^f_2 = y^f + \nu_2$$
$$y^f_3 = y^f + \nu_3$$

6. Calculate the GGD feature a for each potential restored block:

$$b_i = \left(\frac{1}{\sigma_i}\sqrt{\frac{\Gamma(3/\phi)}{\Gamma(1/\phi)}}\right) \quad a_i = \left(\frac{b_i \phi}{2\Gamma(1/\phi)}\right)$$
$$A = \{a_0, a_1, a_2, a_3\}$$

7. Select the restored block that gets the max or min of the GDD feature a:

$$encoder\_flag = 0 \rightarrow \{a_i = \max(A)\}$$
$$encoder\_flag = 1 \rightarrow \{a_i = \min(A)\}$$

$$\gamma = \mathbf{v_i \Phi} \quad \mathbf{y_{resto}} = \mathbf{y} + \gamma$$

Table 4-6: Sparse prediction algorithm at the decoder.

		BD-Rate [3]		
Sequence	Implementation	PSNR	SSIM	VMAF
A2	AV2 + switchable filter	**-1.816**	-0.035	-1.994
	AV2 + SRM	0.487	**-0.337**	**-2.206**
A3	AV2 + switchable filter	**-1.813**	**-0.183**	**-2.310**
	AV2 + SRM	0.794	0.763	-1.642
A4	AV2 + switchable filter	**-2.156**	**-1.890**	**-0.326**
	AV2 + SRM	1.158	1.355	0.730
A5	AV2 + switchable filter	**-0.499**	**-0.627**	**-1.33**
	AV2 + SRM	0.615	-0.174	-1.276
B1	AV2 + switchable filter	**-0.337**	0.025	-1.603
	AV2 + SRM	1.047	**-0.949**	**-2.585**

Table 4-7: BD-Rate (PSNR, VMAF and SSIM).

5 Experimental validation

5.1 Datasets and quality metrics

The standardization group for AV2 recommends a series of raw video sequences with diverse characteristics, including content type, bit-depth, resolution, and color sub-sampling, intending to provide enough scenarios to test new tools against the reference codec. Thus, five test sequences were selected to assess the performance of the sparse in-loop restoration method. The selection was made based on two criteria: 1) Sequences with content proper for All Intra (AI) configuration; 2) Due to the computational cost, we choose sequences with $resolutions < 1920 \times 1080$ and $bit\text{-}depth = 8$. Table **5-1** presents general details of the selected content. In addition, the standardization group defines a group of QP values to evaluate each configuration. In our case, we follow the AI recommendation: $QP = \{85, 110, 135, 160, 185, 210\}$. Raw videos are publicly available and hosted at the open-source platform: Xiph[1]. Figure **5-1** shows examples of frames belonging to the sub-classes A2-A5 and B1.

Class	Sub-class	Resolution	Bit-depth	Total
Natural Videos (A)	$A2$	1920×1080	8	10
	$A3$	1280×720	8	6
	$A4$	640×360	8	6
	$A5$	480×270	8	3
Synthetic (B)	$B1$	1920×1080	8	7

Table 5-1: Selected raw video test sequences

Regarding quality assessment metrics, the standardization group recommends the usage of PSNR per channel (YUV), SSIM [45], and VMAF [2]. Being the less computationally

[1]https://media.xiph.org/video/aomctc/test_set/
[2]https://github.com/Netflix/vmaf

Figure 5-1: Examples of video test sequences (Y plane).

expensive, PSNR is still the most popular metric. However, it is well-known that PSNR correlation with subjective assessments is only partially consistent when evaluating processes such as denoising and restoration. Therefore, PSNR, VMAF, and SSIM are utilized as video quality no-reference assessment metrics.

5.2 Experimental protocol

The experimentation involves four steps:

1. Dictionary selection: assessment of three methodologies, described in sections 3.1.1-3 for establishing the proper dictionary to model the decoding residual, perform training (when applicable) using the sequences A1 and B1 from the database sample and target four block sizes: 8×8, 16×16, 32×32, and 64×64; and ran a series of testing using the content categories A1-A5 and B1 with different block sizes.

2. Sparse in-loop restoration evaluation: compare the performance of PSNR/VMAF/SSIM gain and signaling bits added to the decoder for all the block sizes.

3. Statistical nonzero prediction: evaluate the same three-factor using statistical prediction for the nonzero coefficients in the sparse vector.

4. Sparse in-loop restoration performance: execute a batch of testing for all content categories and compare them against the AV2 restoration tool in terms of PSNR, VMAF, SSIM, and BD-Rate.

The following sections present the result of the steps mentioned above.

5.2.1 Dictionary

Table **5-2** presents the chosen basis for the joint orthogonal approach. DCT refers to the Discrete Cosine, and DWT refers to the Discrete Wavelet transforms. The identity (I) is also included. In addition, we also add the single basis DCT and DWT (no-concatenation). This first approach does not demand training.

Dictionary	Dictionary sizes
$DCT + I$	$64 \times 128, 256 \times 512, 1024 \times 2048$
DCT	$64 \times 64, 256 \times 256, 1024 \times 1024$
$DCT + DWT$	$64 \times 128, 256 \times 512, 1024 \times 2048$
DWT	$64 \times 64, 256 \times 256, 1024 \times 1024$
$DWT + I$	$64 \times 128, 256 \times 512, 1024 \times 2048$

Table 5-2: Joint dictionary basis configuration.

Table **5-3** describes the configuration used for the multi-dictionaries approach. For the training phase, we use the content sequence A2. The original algorithm aims to model the reference frame; in our case, we adapted it to model the residual for three QP levels: 210, 85, and 135. In addition, we defined the first dictionary of the group to be a DCT basis. And, we also utilize different cluster sizes: $K = 10$, $K = 50$, and $K = 100$ to evaluate the performance.

QP	Dictionary sizes	Clusters per Dictionary size
210	$64 \times 64, 256 \times 256, 1024 \times 1024$	$(10 + 1), (50 + 1), (100 + 1)$
135	$64 \times 64, 256 \times 256, 1024 \times 1024$	$(10 + 1), (50 + 1), (100 + 1)$
85	$64 \times 64, 256 \times 256, 1024 \times 1024$	$(10 + 1), (50 + 1), (100 + 1)$

Table 5-3: Multi-dictionaries configuration.

Table **5-4** describes the configuration used for the universal dictionary approach. We use sequence A2 with the associated residual blocks for the training phase for the following QP configurations: 85, 135, and 210. The objective of the universal approach is to build over-complete dictionaries where $n \gg m$. In our case, we used a $4 : 1$ ratio.

Table **5-5** describes the performance, in terms of PSNR, for eight dictionaries configurations, detailed in table **5-6**. Despite the result of the Universal dictionary (Φ_6), the Discrete Cosine Transform (DCT) was selected due to its effect of a high-statistical correlation of the magnitude and location of the nonzero coefficients of the sparse vector. This factor is crucial for prediction on the encoder side, which is a fundamental part of the proposed method.

QP	Dictionary sizes
210	$64 \times 256, 256 \times 1024, 1024 \times 4096$
135	$64 \times 256, 256 \times 1024, 1024 \times 4096$
85	$64 \times 256, 256 \times 1024, 1024 \times 4096$

Table 5-4: Universal dictionary configuration.

Bl	QP	$\|\alpha\|_0$	Ref	Φ_1	Φ_2	Φ_3	Φ_4	Φ_5	Φ_6	Φ_7	Φ_8
16	85	2	45.09	45.62	45.49	45.50	45.28	45.55	45.66	45.56	45.59
16	160	2	34.26	35.89	35.68	35.75	35.10	35.44	36.38	35.00	35.03
16	210	2	28.96	32.45	32.34	32.44	31.26	31.48	34.39	30.62	30.64
16	85	4	45.15	46.11	45.86	45.87	45.53	45.97	46.12	45.91	45.95
16	160	4	34.18	36.73	36.37	36.41	35.27	35.83	37.29	35.20	35.23
16	210	4	28.34	32.74	32.50	32.61	30.66	31.08	34.85	30.10	30.14
16	85	8	45.41	47.38	46.57	46.60	46.16	47.15	47.00	46.71	46.75
16	160	8	34.31	38.13	37.58	37.67	35.96	36.76	38.77	35.90	35.96
16	210	8	27.97	34.18	33.70	33.79	31.06	31.70	36.00	30.43	30.48
8	85	2	44.90	46.35	46.10	46.15	45.56	46.16	46.55	46.39	46.52
8	160	2	34.47	37.94	37.59	37.74	36.60	37.25	39.00	36.88	36.97
8	210	2	28.88	34.78	34.61	34.74	33.14	33.66	37.38	32.93	33.05
8	85	4	44.82	47.45	46.93	47.00	46.03	47.00	47.47	47.24	47.41
8	160	4	34.71	40.00	39.39	39.58	37.65	38.72	41.34	38.19	38.32
8	210	4	28.57	36.49	36.07	36.25	33.28	34.30	39.28	33.43	33.60
8	85	8	45.08	49.75	48.61	48.75	47.16	48.89	49.25	49.08	49.26
8	160	8	34.94	42.81	41.72	41.94	38.62	40.45	43.91	39.88	40.10
8	210	8	28.60	39.54	38.44	38.61	34.15	36.10	41.48	35.31	35.54
Average			36.03	40.24	39.75	39.85	38.25	39.08	**41.23**	38.60	38.70
PSNR gain(%)				11.67	10.32	10.60	6.14	8.46	**14.41**	7.11	7.39

Table 5-5: PSNR after restoration using different dictionaries and raw video sequence A2.

5.3 Sparse in-loop restoration evaluation

Figure **5-2** shows an example of image restoration using sparse representation theory. In this example were utilized block-sizes of 16×16 and a criteria of four nonzero coefficients. We can see a gain of around 0.42 dB. The Orthogonal matching pursuit (OMP) algorithm is used to select the four most relevant atoms from a DCT basis as dictionary.

Figure **5-3** describes the improvement in PSNR for sparse restoration using a block size of 32×32 and 2,4,6,8 and nonzero coefficients. It can be identified that the atoms of the DCT basis are trying to define a shape in some regions. There is also a Block effect, managed by

Dictionary	Description
Φ_1	$DCT + I$
Φ_2	DCT
Φ_3	$DCT + DWT$
Φ_4	DWT
Φ_5	$DWT + I$
Φ_6	Universal
Φ_7	Multi-dictionaries (10 clusters)
Φ_8	Multi-dictionaries(20 clusters)

Table 5-6: Evaluated dictionaries.

overlapping blocks, as depicted in **5-4**.

5.4 Statistical nonzero prediction

In addition to the result presented in Chapter 4, Figures 5-10 describe the histogram of the magnitude of the AC nonzero coefficients for $QP \in \{85, 110, 135, 160, 185, 210\}$. Figure **5-16** presents the distribution for QP=85. On the left side is the distribution for DC nonzeros, as we notice the low amount of data (peak approx. 10), which led us to discard DC for QP=85. In others words, for this QP, we assume all the coefficients of the sparse vector are all in AC. This assumption is used across the development of the model in chapter 4. For QP=85, the histogram tends to a Laplace distribution which aligns with the lossless compression for this quality level, and consequently, the coefficients are concentrated around μ. For the remaining QPs, the DC nonzero coefficient was modeled as a Gamma distribution; for AC, it was defined as Gaussian. We found highly accurate results, in terms of visual quality, for $QP < 185$. In the scenario of QP $\in \{185, 210\}$, we discovered that the images tend to create a blocking effect when the Gaussian prediction outputs peak values (figure **5-13**. We experimentally encountered that reducing a factor of 0.35 and 0.6 for QP $\in \{185, 210\}$ eliminates this artifact **5-14**. The blocking impairment is only perceived during subjective evaluation, even though all metrics, including PSNR, SSIM, and VMAF, still show gains.

Regarding the Gaussian estimation of the AC coefficient, we obtained a performance where nearly 1% of PSNR quality is sacrificed compared to the non-prediction approach (classical),

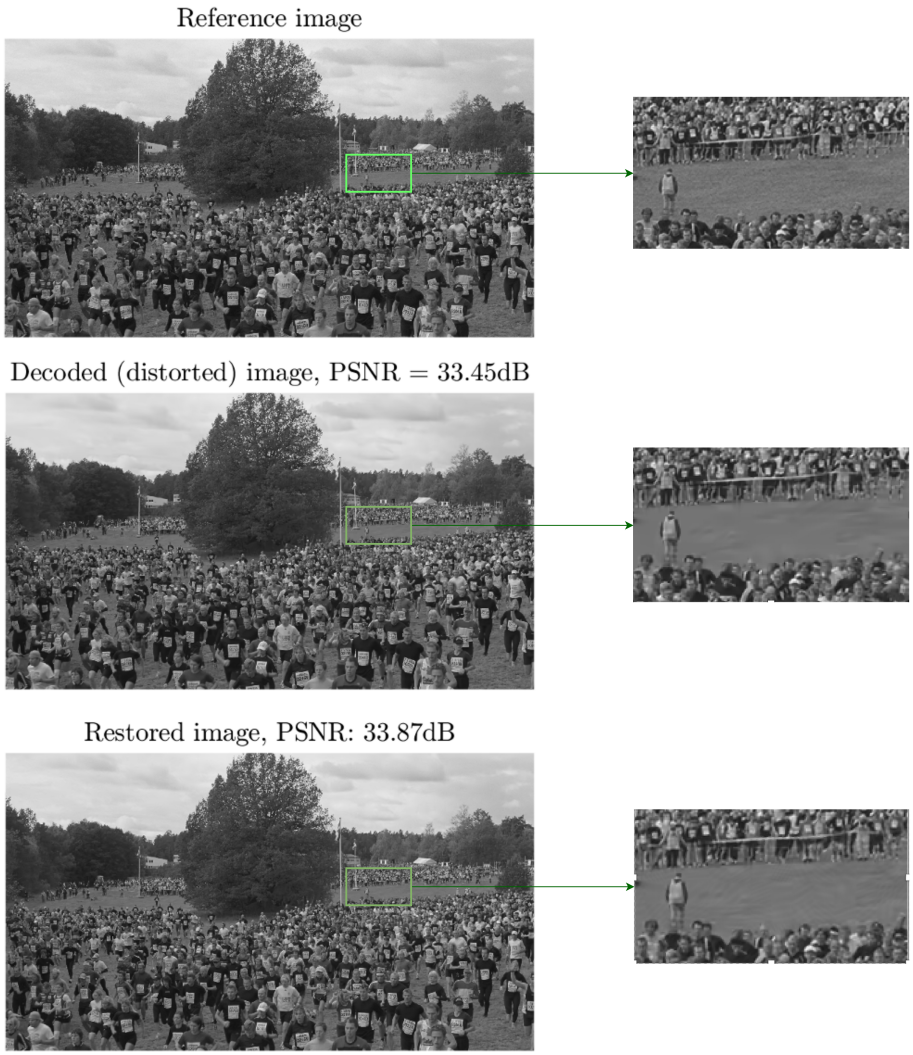

Figure 5-2: Frame restoration QP=135.

figure **5-11** and **5-12** illustrate that. That means Gaussian estimation reduces the gain while eliminating the need to share the nonzero coefficients (a double-precision number of 2 bytes). This prediction makes our Sparse restoration method feasible for real implementations.

Regarding the position of the non-zero coefficient, we experimentally encountered a concentration around bands and highly spread for lossless QP, i.e., 85 when the DCT basis is used as a dictionary. Figure **5-15** compares two QPs using DCT and a Universal dictionary. The DCT is concentrated, and more predictable, which supports the proper estimation of the decoding residual discussed and presented in Chapter 4.

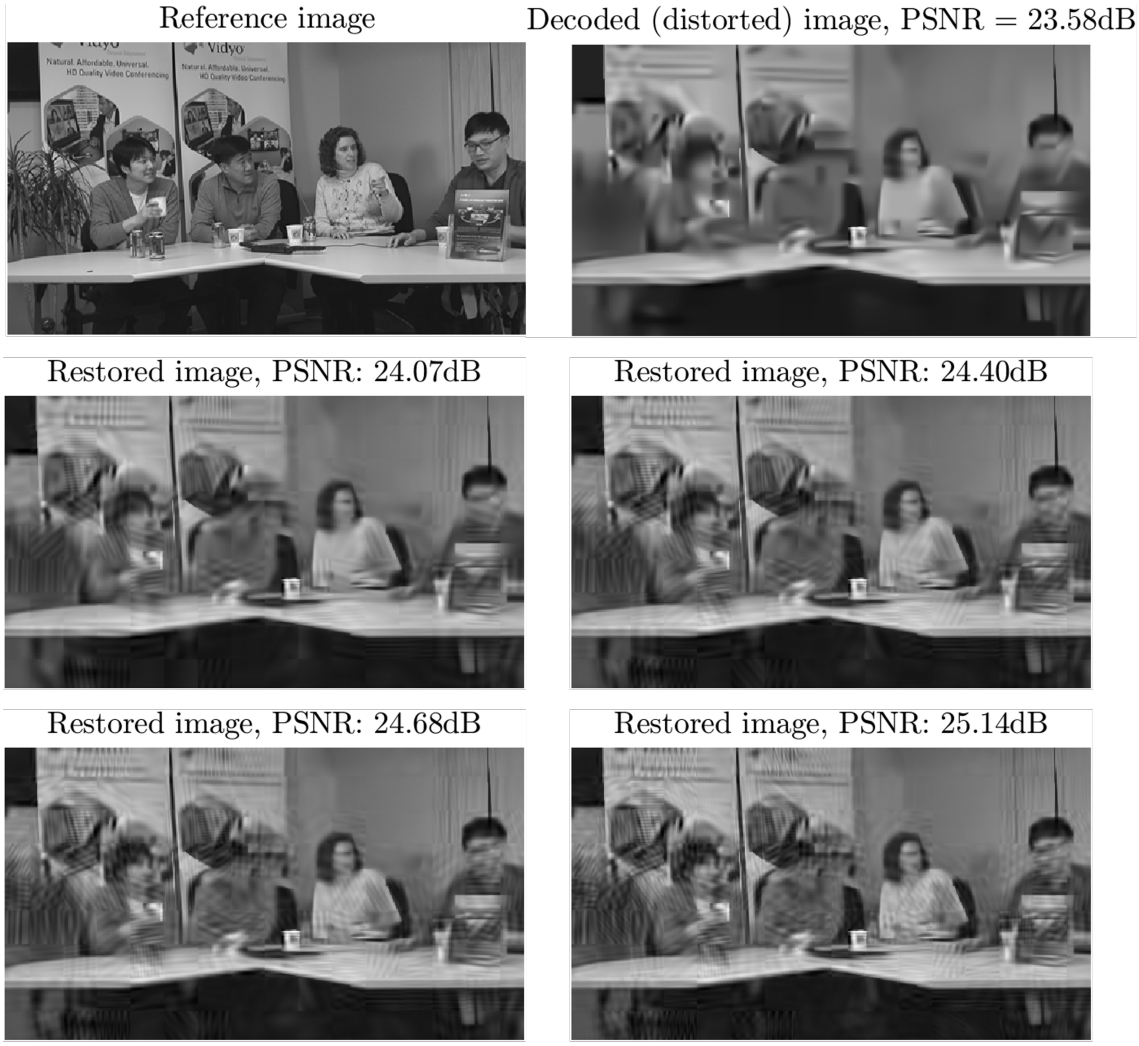

Figure 5-3: Frame restoration for QP=185, block-size= 32×32, nonzero $\in \{2, 4, 6, 8\}$, overlapping factor=1.

5.5 Sparse in-loop restoration performance

In addition to the results presented in Chapter 3, we evaluate two GDD features. The first is a, described and used in Chapter 3 and ζ, the frequency factor defined as $\zeta = \sigma^2/\mu$. We executed tests where: 1) Only AC coefficients were considered, and DC was skipped. 2) DC coefficients are modeled when the variance of the distorted block y is less than 1. And 3). Includes always the DC coefficient as part of the potential options for modeling the decoding residual. Table **5-7** presents the results.

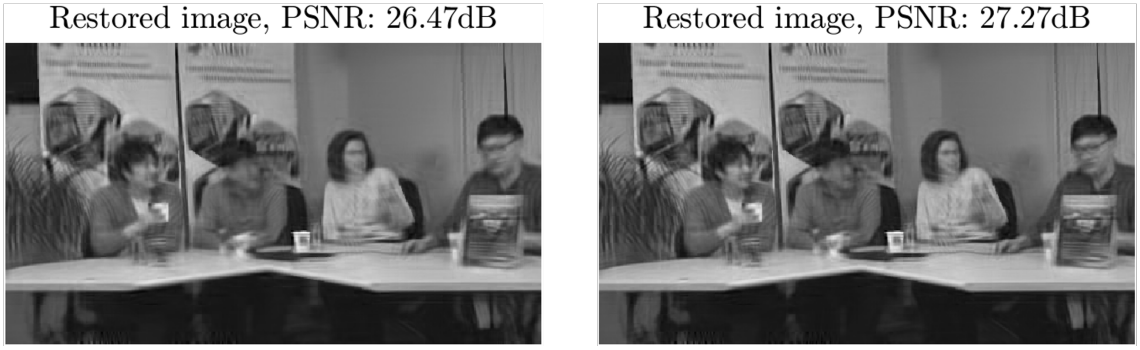

Figure 5-4: Frame restoration QP=185, block-size $=16\times16$, nonzeros $\in\{2,4\}$, overlapping factor=4.

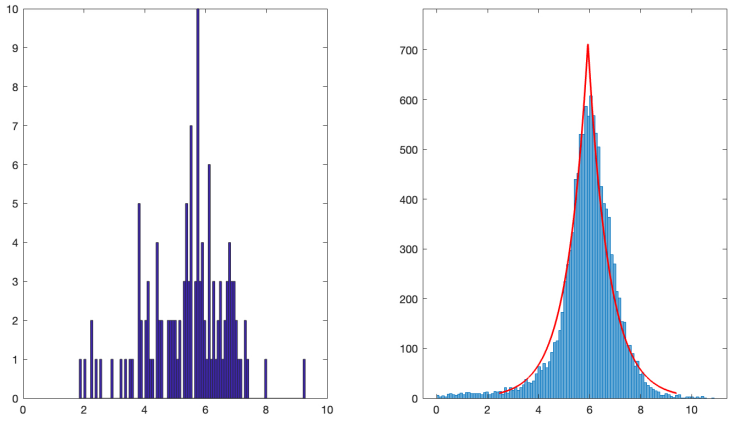

Figure 5-5: $QP=85$, Laplace PDF for AC (right) and few DC coefficients (left).

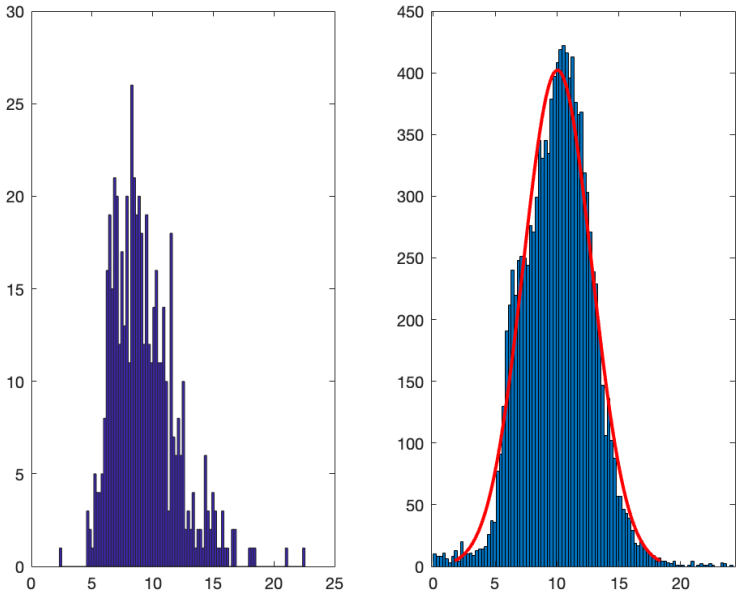

Figure 5-6: $QP = 110$, Gamma PDF for DC (right) and Gaussian PDF for DC (left).

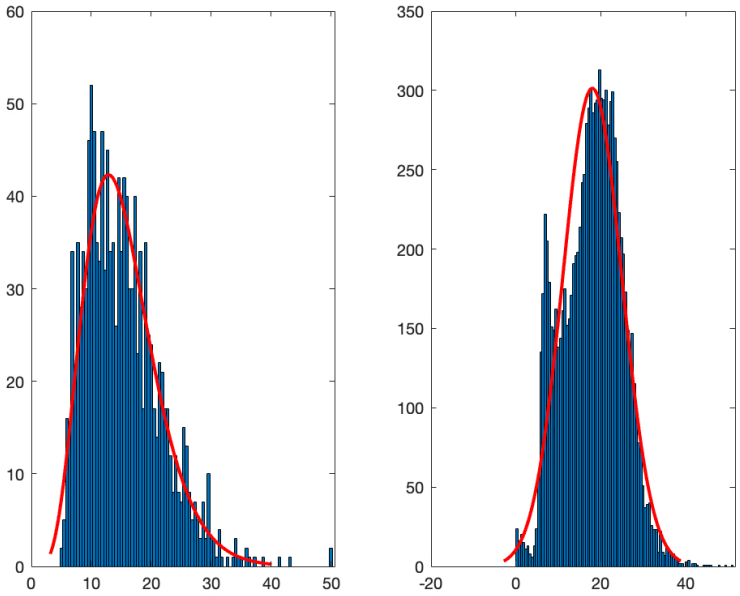

Figure 5-7: $QP = 135$, Gamma PDF for DC (right) and Gaussian PDF for DC (left).

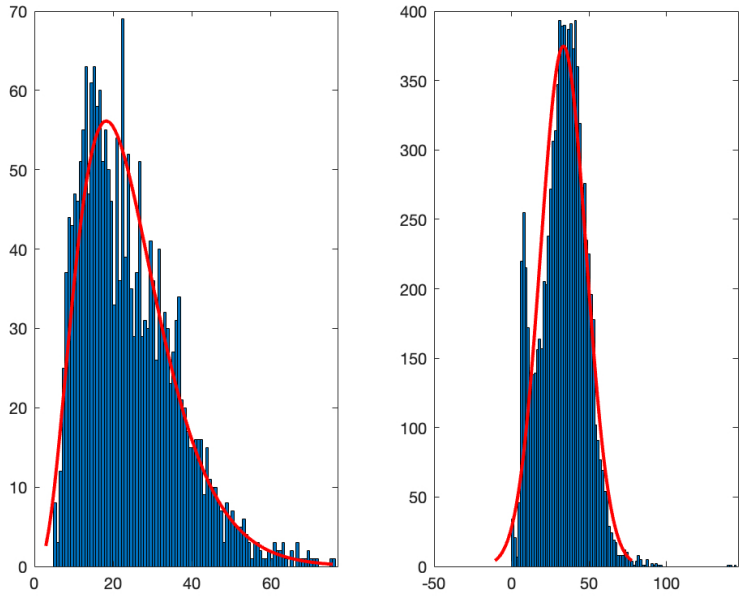

Figure 5-8: $QP = 160$, Gamma PDF for DC (right) and Gaussian PDF for DC (left).

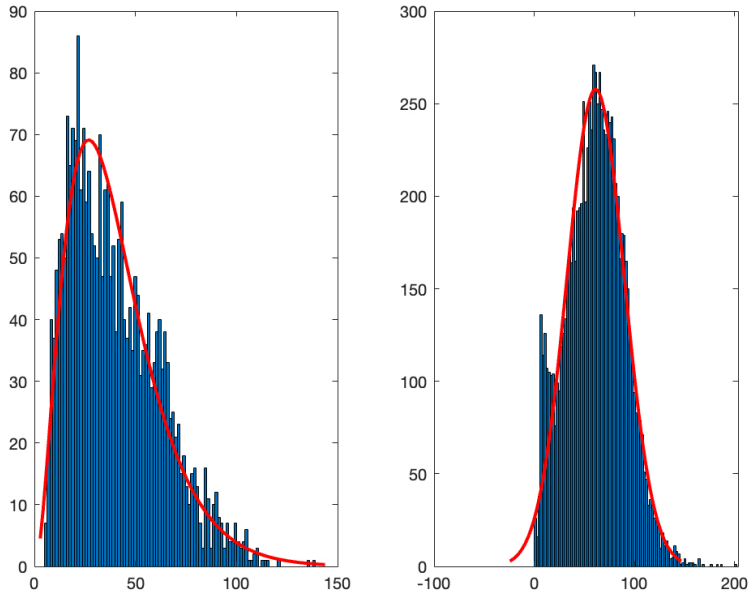

Figure 5-9: $QP = 185$, Gamma PDF for DC (right) and Gaussian PDF for DC (left)

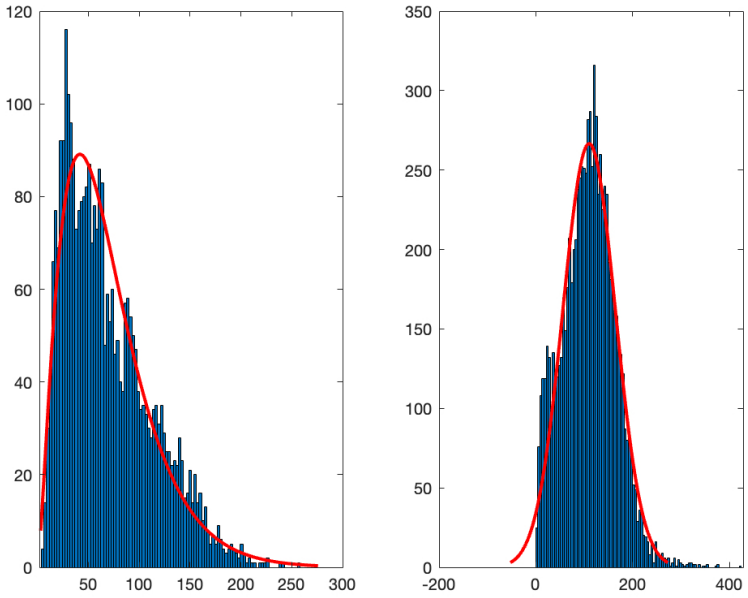

Figure 5-10: $QP = 210$, Gamma PDF for DC (right) and Gaussian PDF for DC (left).

Figure 5-11: Sparse coefficient estimation for $QP = 210$.

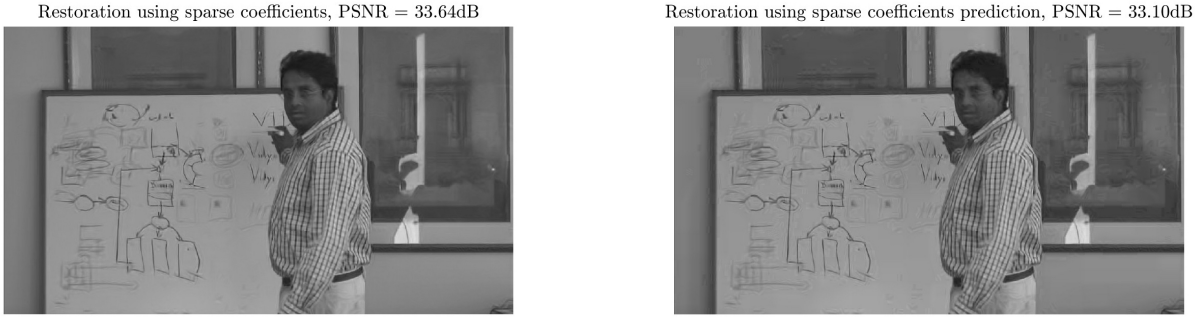

Figure 5-12: Sparse coefficient estimation for $QP = 185$.

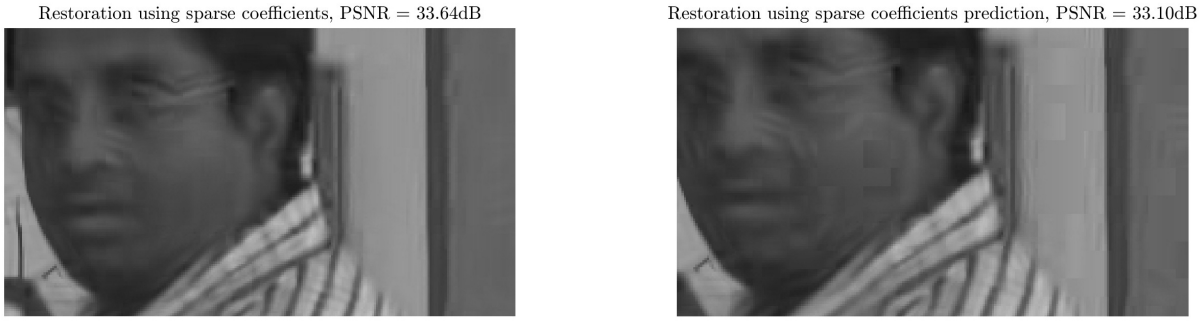

Figure 5-13: Sparse coefficient estimation for $QP = 185$ and blocking effect.

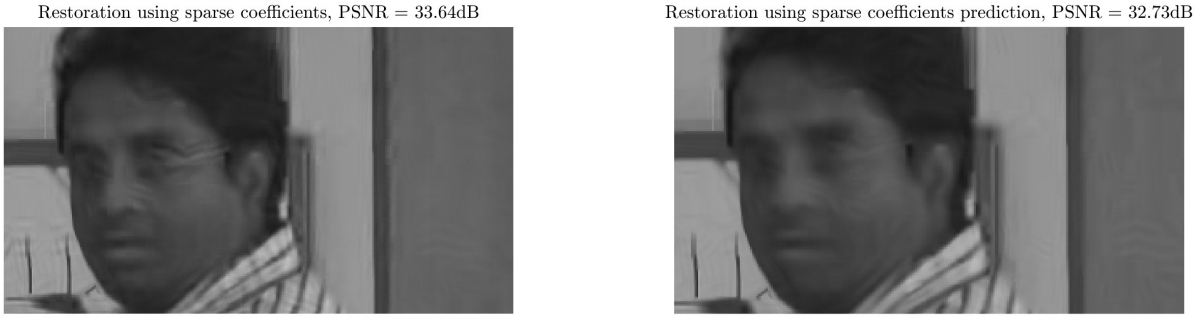

Figure 5-14: Sparse coefficient estimation for $QP = 185$ and blocking effect mitigation.

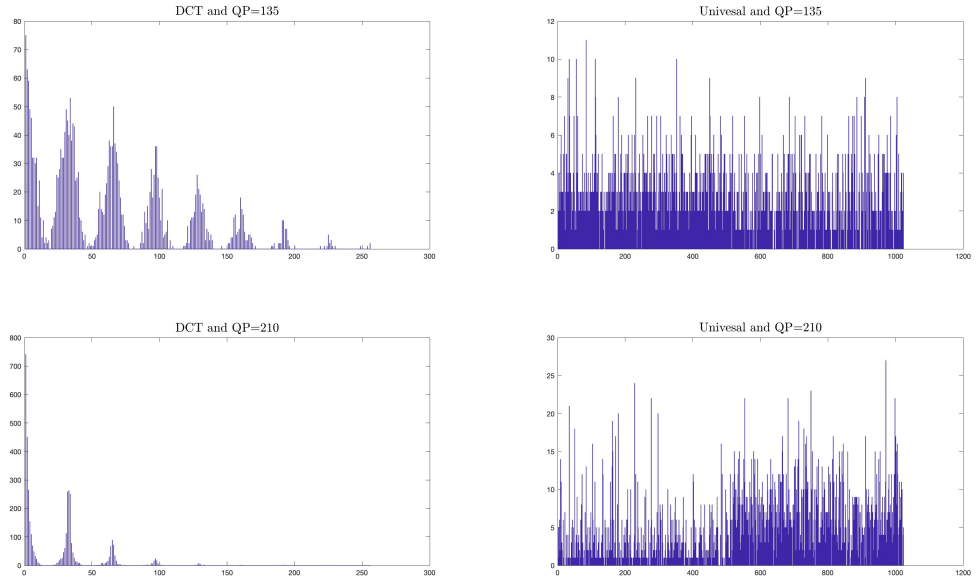

Figure 5-15: Distribution of nonzero coefficients across the sparse vector.

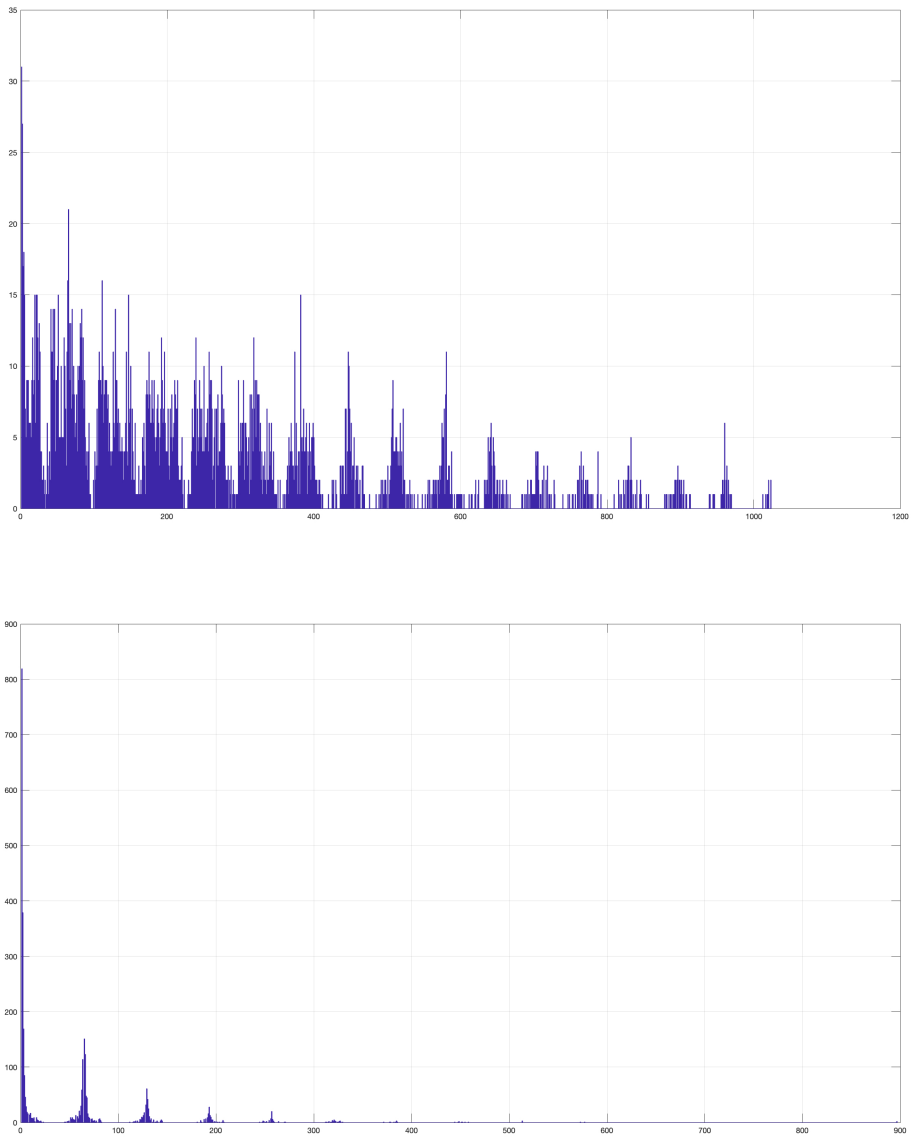

Figure 5-16: Distribution of sparse nonzero coefficient position $QP = 85$ (top) and $QP = 210$ (bottom).

			Accuracy		Gain	
QP	Description	Criteria	SSIM	PSNR	SSIM	PSNR
210	Only AC	a	49.65	46.55	8.84	1.30
		ζ	48.12	48.82	10.45	1.27
210	DC+AC	a	51.93	57.54	9.71	2.77
	$var(y) < 1$	ζ	48.96	47.06	10.97	2.39
210	DC+AC	a	53.42	61.97	6.21	2.78
	$var(y) > 0$	ζ	49.52	48.57	7.90	2.89
160	Only AC	a	50.02	46.72	1.82	0.37
		ζ	48.63	44.20	1.89	0.38
160	DC+AC	a	47.62	54.91	1.55	0.66
	$var(y) < 1$	ζ	43.47	43.58	1.80	0.56
160	DC+AC	a	51.40	59.37	1.01	0.62
	$var(y) > 0$	ζ	47.63	47.77	1.16	0.62
85	Only AC	a	39.24	46.40	0.11	0.04
		ζ	42.63	42.42	0.12	0.04
85	AC+DC	a	40.27	49.39	0.11	0.04
	$var(y) < 1$	ζ	44.37	43.31	0.09	0.03
85	AC+DC	a	44.00	59.30	0.08	0.04
	$var(y) > 0$	ζ	46.99	42.71	0.07	0.04
Overall		a	**47.50**	**53.57**	**3.27**	**0.95**
		ζ	46.70	45.38	3.82	0.91

Table 5-7: Prediction accuracy and gain against PSNR and SSIM.

6 Conclusions and future works

- Sparse representation is, without a doubt, an efficient approach for image restoration tasks. Regarding the video coding in-loop restoration scenario, the critical challenge is eliminating the information required to transfer between the encoder and decoder to represent the nonzero coefficients. However, moving the high-intensive task to the decoder is not an option, considering the real-time exigency during the decoding process. Therefore, we developed a hybrid approach where most of the required information is predicted in the decoder, and only a guiding bit $encoder\text{-}flag$ was required. The reason for utilizing a guiding bit is the poor precision of predicting if a block, i.e., 32×32, needs to collapse or expand in terms of the GGD. We ran experiments using various methods, including CNN, and non-reference quality metrics, such as NIQE and PIQE; the accuracy was not higher than 25%. It is difficult to predict because the method tries to be as closer as possible to a block, using PSNR and SSIM as error metrics which still need to be fully correlated with non-reference metrics at the block level.

- Our result points to the following research where a highly correlated video quality metric at the DCT level, such as BLIND-II [38], can be adapted to run in real-time and allows blind-restoration using the Sparse estimation algorithm presented in this document. A critical factor that could accelerate the development of restoration models in this research line is defining quality at the block level and correlating with the whole frame.

- Deep-learning tools for in-loop restoration are the right direction to tackle this problem. However, those methods must incorporate non-reference quality metrics, as explained before, to be able to run in parallel, which means restoring at the block level and achieving efficiency at the whole frame. In addition, this should exploit frequency features in order to reduce computational cost and time during the decoding process.

7 Bibliography

[1] *Sparse Representations*. 2009

[2] Ahmed, N. ; Natarajan, T. ; Rao, K.R.: Discrete Cosine Transform. En: *IEEE Transactions on Computers* C-23 (1974), Nr. 1, p. 90–93

[3] Antsiferova, Anastasia ; Lavrushkin, Sergey ; Smirnov, Maksim ; Gushchin, Aleksandr ; Vatolin, Dmitriy S. ; Kulikov, Dmitriy. *Video compression dataset and benchmark of learning-based video-quality metrics*. 2022

[4] Barman, Nabajeet ; Martini, Maria G. ; Reznik, Yuriy: Revisiting Bjontegaard Delta Bitrate (BD-BR) Computation for Codec Compression Efficiency Comparison. En: *Proceedings of the 1st Mile-High Video Conference*. New York, NY, USA : Association for Computing Machinery, 2022 (MHV '22). – ISBN 9781450392228, p. 113–114

[5] Cai, T. T. ; Wang, Lie: Orthogonal matching pursuit for sparse signal recovery with noise. En: *IEEE Transactions on Information Theory* 57 (2011), 7, p. 4680–4688. – ISSN 00189448

[6] Chen, Ching-Yeh ; Tsai, Chia-Yang ; Huang, Yu-Wen ; Yamakage, Tomoo ; Chong, In S. ; Fu, Chih-Ming ; Itoh, Takayuki ; Watanabe, Takashi ; Chujoh, Takeshi ; Karczewicz, Marta ; Lei, Shaw-Min: The adaptive loop filtering techniques in the HEVC standard. En: Tescher, Andrew G. (Ed.): *Applications of Digital Image Processing XXXV* Vol. 8499, 2012, p. 849913

[7] Dai, Yuanying ; Liu, Dong ; Wu, Feng: A Convolutional Neural Network Approach for Post-Processing in HEVC Intra Coding.

[8] Ding, Dandan ; Chen, Guangyao ; Mukherjee, Debargha ; Joshi, Urvang ; Chen, Yue: A CNN-based In-loop Filtering Approach for AV1 Video Codec. En: *2019 Picture Coding Symposium (PCS)*, 2019, p. 1–5

[9] DIng, Dandan ; Chen, Guangyao ; Mukherjee, Debargha ; Joshi, Urvang ; Chen, Yue: A progressive CNN in-loop filtering approach for inter frame coding. En: *2019 Picture Coding Symposium, PCS 2019* (2019). ISBN 9781728147048

[10] Dong, Junshuo ; Wu, Lingda: Comparison and Simulation Study of the Sparse Representation Matching Pursuit Algorithm and the Orthogonal Matching Pursuit Algorithm. (2021), p. 317–320

[11] Dong, Weisheng ; Shi, Guangming ; Li, Xin: Image deblurring with low-rank approximation structured sparse representation. En: *2012 Conference Handbook - Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, APSIPA ASC 2012* (2012), p. 14–18. ISBN 9780615700502

[12] Dong, Weisheng ; Zhang, Lei ; Shi, Guangming ; Li, Xin: Nonlocally centralized sparse representation for image restoration. En: *IEEE Transactions on Image Processing* 22 (2013), Nr. 4, p. 1620–1630. – ISSN 10577149

[13] Dong, Weisheng ; Zhang, Lei ; Shi, Guangming ; Li, Xin: Nonlocally centralized sparse representation for image restoration. En: *IEEE Transactions on Image Processing* 22 (2013), Nr. 4, p. 1620–1630. – ISSN 10577149

[14] Dong, Weisheng ; Zhang, Lei ; Shi, Guangming ; Wu, Xiaolin: Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. En: *IEEE Transactions on Image Processing* 20 (2011), Nr. 7, p. 1838–1857. – ISSN 10577149

[15] Donoho, David L.: Compressed sensing. En: *IEEE Transactions on Information Theory* 52 (2006), Nr. 4, p. 1289–1306. – ISSN 00189448

[16] Donoho, David L. ; Elad, Michael: Optimally sparse representation in general (nonorthogonal) dictionaries via 1 minimization. En: *PNAS March* 4 (2003), p. 2197–2202

[17] Ebadi, Salehe E. ; Ones, Valia G. ; Izquierdo, Ebroul: UHD Video Super-Resolution Using Low-Rank and Sparse Decomposition. En: *Proceedings - 2017 IEEE International Conference on Computer Vision Workshops, ICCVW 2017* 2018-Janua (2017), p. 1889–1897. ISBN 9781538610343

[18] Ekstrom, Michael P.: Realizable Wiener Filtering in Two Dimensions. En: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 30 (1982), p. 31–40. – ISSN 00963518

[19] Elad, Michael ; Aharon, Michal: Image denoising via learned dictionaries and sparse representation. En: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 1 (2006), p. 895–900. – ISBN 0769525970

[20] Elad, Michael ; Bruckstein, Alfred M.: A generalized uncertainty principle and sparse representation in pairs of bases. En: *IEEE Transactions on Information Theory* 48 (2002), 9, p. 2558–2567. – ISSN 00189448

[21] Han, Jingning ; Li, Bohan ; Mukherjee, Debargha ; Chiang, Ching-Han ; Chen, Cheng ; Su, Hui ; Parker, Sarah ; Joshi, Urvang ; Chen, Yue ; Wang, Yunqing ; Wilkins, Paul ; Xu, Yaowu ; Bankoski, James: A Technical Overview of AV1. (2020), 8

[22] Han, Jingning ; Xu, Yaowu ; Mukherjee, Debargha: A butterfly structured design of the hybrid transform coding scheme. (2013), p. 17–20

[23] Hastie, Trevor ; Martin, Robert T. ; Hastie, Wainwright ; Tibshirani, * ; Wainwright, *. *Statistical Learning with Sparsity The Lasso and Generalizations Statistical Learning with Sparsity*

[24] Ji, Hui ; Huang, Sibin ; Shen, Zuowei ; Xu, Yuhong: Robust Video Restoration by Joint Sparse and Low Rank Matrix Approximation. En: *SIAM Journal on Imaging Sciences* 4 (2011), Nr. 4, p. 1122–1142

[25] Jia, Chuanmin ; Wang, Shiqi ; Zhang, Xinfeng ; Wang, Shanshe ; Liu, Jiaying ; Pu, Shiliang ; Ma, Siwei: Content-Aware Convolutional Neural Network for In-Loop Filtering in High Efficiency Video Coding. En: *IEEE Transactions on Image Processing* 28 (2019), p. 3343–3356. – ISSN 19410042

[26] Kato, Toshiyuki ; Hino, Hideitsu ; Murata, Noboru: Sparse Coding Approach for Multi-Frame Image Super Resolution. (2014), p. 1–20

[27] Kim, Jiwon ; Lee, Jung K. ; Lee, Kyoung M.: Accurate Image Super-Resolution Using Very Deep Convolutional Networks. En: *CoRR* abs/1511.04587 (2015)

[28] Kong, Lingyi ; Ding, Dandan ; Liu, Fuchang ; Mukherjee, Debargha ; Joshi, Urvang ; Chen, Yue: Guided CNN Restoration with Explicitly Signaled Linear Combination. En: *Proceedings - International Conference on Image Processing, ICIP* 2020-Octob (2020), p. 3379–3383. – ISBN 9781728163956

[29] Lin, Liqun ; Yu, Shiqi ; Zhao, Tiesong ; Wang, Zhou: PEA265: Perceptual Assessment of Video Compression Artifacts. (2019)

[30] Maćkiewicz, Andrzej ; Ratajczak, Waldemar: Principal Components Analysis (PCA). En: *Computers & Geosciences* 19 (1993), p. 303–342

[31] Mairal, Julien ; Elad, Michael ; Sapiro, Guillermo: Sparse representation for color image restoration. En: *IEEE Transactions on Image Processing* 17 (2008), p. 53–69. – ISSN 10577149

[32] Mairal, Julien ; Sapiro, Guillermo ; Elad, Michael: Learning multiscale sparse representations for image and video restoration. En: *Multiscale Modeling and Simulation* 7 (2008), Nr. 1, p. 214–241. – ISSN 15403467

[33] Moorthy, Anush K. ; Bovik, Alan C. *STATISTICS OF NATURAL IMAGE DISTORTIONS*

[34] Mukherjee, Debargha ; Han, Jingning ; Bankoski, Jim ; Bultje, Ronald ; Grange, Adrian ; Koleszar, John ; Wilkins, Paul ; Xu, Yaowu: A Technical Overview of VP9 – The Latest Open-Source Video Codec. (2013), p. 1–17

[35] O'Shea, Keiron ; Nash, Ryan: An Introduction to Convolutional Neural Networks. En: *CoRR* abs/1511.08458 (2015)

[36] Oxford: *A Dictionary of Statistics*. Oxford University Press, 2014. – ISBN 9780191758317

[37] Reininger, Randall C. ; Gibson, Jerry D.: Distributions of the Two-Dimensional DCT Coefficients for Images. En: *IEEE Transactions on Communications* 31 (1983), p. 835–839. – ISSN 00906778

[38] Saad, Michele A. ; Bovik, Alan C. ; Charrier, Christophe: DCT statistics model-based blind image quality assessment, 2011. – ISBN 9781457713033, p. 3093–3096

[39] Sankaraiah, Yediga R. ; Varadarajan, Sourirajan: An effective image deblurring scheme using cluster based sparse representation. En: *ASEAN Engineering Journal* 11 (2021), Nr. 4, p. 16–28. – ISSN 25869159

[40] Scetbon, Meyer ; Elad, Michael ; Milanfar, Peyman: Deep K-SVD denoising. En: *IEEE Transactions on Image Processing* 30 (2021), Nr. 8, p. 5944–5955. – ISSN 19410042

[41] Schneider, Jens ; Sauer, Johannes ; Wien, Mathias: RDPlot – An Evaluation Tool for Video Coding Simulations. En: *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 2021, p. 1–1

[42] Segall, C A. ; Katsaggelos, Aggelos K. ; Molina, Rafael: Chapter 11 Super-resolution from compressed video. En: *Book* (2001), p. 1–32

[43] Siekmann, Mischa ; Bosse, Sebastian ; Schwarz, Heiko ; Wiegand, Thomas: SEPARABLE WIENER FILTER BASED ADAPTIVE IN-LOOP FILTER FOR VIDEO CODING Image Processing Department Fraunhofer Institute for Telecommunications – Heinrich Hertz Institute Image Communication Chair Department of Telecommunication Systems Technical Universit. (2010), p. 70–73. ISBN 9781424471355

[44] Valin, Jean-Marc: The Daala Directional Deringing Filter. En: *CoRR* abs/1602.05975 (2016)

[45] Wang, Z. ; Simoncelli, E.P. ; Bovik, A.C.: Multiscale structural similarity for image quality assessment. En: *The Thrity-Seventh Asilomar Conference on Signals, Systems and Computers, 2003* Vol. 2, 2003, p. 1398–1402 Vol.2

[46] Wang, Zhou ; Bovik, A.C. ; Sheikh, H.R. ; Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. En: *IEEE Transactions on Image Processing* 13 (2004), Nr. 4, p. 600–612

[47] Wiener, Norbert: *Extrapolation, interpolation, and smoothing of stationary time series with engineering applications*. 1964. – ISBN 9780262730051

[48] Y., Dodge: *Gamma Distribution*. New York, NY : Springer New York, 2008. – 215–216 p.. – ISBN 978–0–387–32833–1

[49] Yang, Jianchao ; Wright, John ; Huang, Thomas S. ; Ma, Yi: Image super-resolution via sparse representation. En: *IEEE Transactions on Image Processing* 19 (2010), Nr. 11, p. 2861–2873. – ISSN 10577149

[50] Zhu, Shujin ; Yu, Zekuan: Self-guided filter for image denoising. En: *IET Image Processing* 14 (2020), Nr. 11, p. 2561–2566