

## Article

# A Novel Online Correlation Noise Model Based on Band Coefficients Mean to Achieve Low Computational and Coding-Efficient Distributed Video Codec

Shahzad Khursheed <sup>1,\*</sup> , Nasreen Badruddin <sup>1,\*</sup> , Varun Jeoti <sup>2</sup> and Manzoor Ahmed Hashmani <sup>3</sup> 

<sup>1</sup> Department of Electrical and Electronic Engineering, Institute of Health and Analytics, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia

<sup>2</sup> Faculty of Technical Sciences, University of Novi Sad, 21000 Novi Sad, Serbia; varunjeoti@uns.ac.rs

<sup>3</sup> High Performance Cloud Computing Center (HPC3), Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Seri Iskandar 32610, Malaysia; manzoor.hashmani@utp.edu.my

\* Correspondence: shahzad\_17005202@utp.edu.my (S.K.); nasreen.b@utp.edu.my (N.B.)

**Abstract:** Distributed video coding (DVC) is a novel coding paradigm that offers low computational encoding relative to conventional video-coding framework at the expense of high-decoding computational complexity. The challenging part of this video-coding framework is achieving better rate-distortion (RD) compared with conventional codec performance. A suitable and accurate correlation noise model (CNM) is crucial in improving the RD performance by achieving high coding efficiency and making decoding less computationally demanding. Since the correlation is nonstationary and time-variant and can vary from frame to frame, offline CNM estimation is not feasible for practical applications and real-time decoding. An online CNM may be the solution to this problem. In DVC, neither Wyner–Ziv frame (WZF) nor estimated side information (SI) of the corresponding WZF is available at the encoder. Therefore, online estimation of the CNM and its parameters can be quite challenging. The contribution of this research work is a novel online CNM which is computed by taking the mean of each transformed coefficient band and deployed for two different codecs. Our proposed codec, DIVCOM, which stands for “Distributed Video Coding with Online Band Mean Correlation Noise Model”, outperforms the existing baseline codec, DISCOVER (DIS), in both coding efficiency and peak signal-to-noise ratio (PSNR). The DIVCOM codec achieves coding efficiency of up to 8.05 kbps, and PSNR ranges from 0.0245 dB to 0.18 dB. An extended version of DIVCOM incorporating phase-based side information called PDIVCOM achieves coding efficiency up to 10.9 kbps, and PSNR ranges from 0.019 to 0.17 dB compared to DIS.

**Keywords:** distributed video coding; online correlation noise model; online correlation noise modeling for DVC applications; CNM for coding efficient



**Citation:** Khursheed, S.; Badruddin, N.; Jeoti, V.; Hashmani, M.A. A Novel Online Correlation Noise Model Based on Band Coefficients Mean to Achieve Low Computational and Coding-Efficient Distributed Video Codec. *Appl. Sci.* **2022**, *12*, 6505. <https://doi.org/10.3390/app12136505>

Academic Editors: Marco Castellani and Francesco Zirilli

Received: 29 April 2022

Accepted: 8 June 2022

Published: 27 June 2022

**Publisher’s Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Many upstream applications of wireless video sensor networks (WVSN), such as multimedia sensor networks, real-time wireless video surveillance, environment monitoring, medical monitoring, and internet of things (IoT), have emerged over recent time [1–4]. The battery size and life and other limitations of the transmitting device are of paramount importance in such applications [5]. For such low-resource upstream applications, low computational complexity at the encoder is desirable [6]. One of the video coding paradigms that support these requirements is distributed video coding (DVC) [7]. This coding paradigm redistributes the coding complexity and shifts most of the computation to the decoder by exploiting the source statistics at the decoder [8] and making the encoder computationally light [9]. However, this new coding design is not fully developed and is getting more attention from the research community to improve its performance compared with conventional codecs for real-time and resource-constrained applications [10,11].

DVC is based on the Slepian–Wolf (SW) [12] and Wyner–Ziv (WZ) [13] theorems of Information Theory. According to the SW theorem, if two correlated sources are encoded separately and decoded jointly, then the minimum coding rate we can achieve is the same as that of joint encoding and decoding. Wyner and Ziv obtained similar bounds for lossy coding with the presence of decoder side information (SI). The two major categories of DVC are block-based and frame-based DVC frameworks. Most of the available DVC codecs in literature are based on the frame-based Stanford architecture [14], which can then be divided into two types—pixel domain WZ (PDWZ) coding and transform domain WZ (TDWZ) coding. TDWZ coding is widely used in the literature due to its coding efficiency, and the DISCOVER codec (DIS) [10] is considered the state-of-the-art codec based on this architecture [14].

In TDWZ DVC schemes, the video is split into frames, and these frames are organized into Group of Pictures (GOP) of sizes 2, 4, or 8. For every GOP, the first frame is the key-frame (KF) and encoded with intra-predictive coding (H.264 intra-encoder). The rest of the GOP frames, called Wyner–Ziv frames (WZFs), are encoded with WZ coding. The WZF is first split into  $4 \times 4$  blocks, and discrete cosine transform (DCT) is applied on each block of the frame. The corresponding coefficients from each block are organized into 16 coefficient bands. Each coefficient band is quantized to the desired quality. For each band, the bit-planes are extracted after quantization and channel encoded (turbo or LDPC) for parity bit generation. These parity bits are stored in a buffer and transmitted on the decoder's request through a feedback channel. At the decoder, the first KF is decoded. Then the SI, which is an approximated replica of the current WZF, is estimated with motion-compensated interpolation or extrapolation of the previous and subsequent decoded frames. One challenge in DVC is to find a channel noise model (CNM) that accurately models the statistical dependency in the form of virtual channel correlation noise, between the current WZF and its corresponding SI. A more accurate model will lead to improved performance in coding efficiency and rate distortion (RD).

In our previous research work [15], efforts were made to develop a phase-based frame interpolation algorithm that can effectively generate the SI faster and with lower computational complexity compared to the baseline codec, DIS. In this paper, however, our main objective is to design an online CNM framework that can be used to develop a full DVC codec that has less decoding computational complexity and can compete with DIS in coding efficiency and RD performance. In most DVC codecs, the computation of error distribution between the original WZF and corresponding SI is challenging due to the unavailability of SI at the encoder and the original WZF at the decoder. Therefore, in this paper, we attempt to design a framework for the online estimation of CNM for the Laplacian distribution parameter while considering the mean of the coefficient band. The main contributions of our research work are the following:

1. Designed a framework for an online correlation noise model (CNM) which was implemented in the DIS codec. This implementation is called DIVCOM, which stands for Distributed Video Coding with Online Band Mean Correlation Noise Model.
2. Implemented the online CNM in the codec developed in a previous work of the authors and presented in [15]. This implementation is called PDIVCOM, which stands for Phase-based Distributed Video Coding with Online Band Mean Correlation Noise Model.
3. Evaluated and compared the performance of the codecs with the proposed online CNM.

The rest of the paper is organized as follows: related work is covered in Section 2, followed by a comprehensive explanation of the DVC framework with the proposed online correlation noise model (CNM) estimation, and its mathematical formulation in Section 3. The results are presented and elaborated in Section 4, followed by a conclusion and future research work in Section 5.

## 2. Related Work

The estimation of an accurate CNM between the correlated sources either at encoder or decoder with fully unknown or partially known data [16,17] is a key problem in DVC. Therefore, a lot of literature is devoted to the estimation of the CNM, but most of them follow the unrealistic offline assumption [11]. In DVC, the overall coding efficiency and RD performance of codec improved with an accurate estimation of parameters of the CNM.

In the DVC codec, neither WZF nor the estimated SI of corresponding WZF is available at the decoder and encoder, respectively. Therefore, offline approaches are adopted to calculate the CNM parameter through the distribution model by assuming either SI is available at the encoder or the actual WZF is available at the decoder. In this approach, the parameter is calculated offline for the whole video sequence and later used in the decoder. This is one of the biggest hurdles in the practical implementation of DVC because offline estimation of the CNM does not perform well for all types of motion videos. Furthermore, the correlation noise is not stationary, and its statistics vary from frame to frame. Therefore, the estimation of CNM parameters for different sequences through the offline CNM is not feasible to achieve coding efficiency and high RD performance. This is because when it is implemented at the decoder, it does not exploit the variability of CNM from frame to frame for actual motion. Furthermore, this is a complex task since the original information is not available in the decoder, and the SI quality varies throughout the sequence. However, if the model accurately describes the WZF and SI relationships, the coding performance is high and vice-versa. A Laplacian distribution model is applied in most architectures because of its excellent trade-off between model accuracy and complexity.

Hence, the focus has been put on online CNM parameter estimation without having access to the original WZF. The authors of [18] proposed several online correlation noise schemes for pixel-based coding at different granularity levels (frame, block, and pixel) by exploiting the temporal correlation between the decoded KF and estimated SI. In an extended work [19], the online CNM is adopted for the transform domain at the band level and coefficient level. Enhancements were made in [20] to the codec in [19] with the introduction of cross-band correlation by calculating residuals between bands, and a band classification map, which is updated after the successful decoding of each band. With cross-band correlation, the classification map of the current band is used to estimate the classification map of the next band. With this estimated cross-band classification of each band, the CNM parameter is calculated. In [21], the correlation noise parameter estimation is done at the DCT band level, where SI is refined after the decoding of each bit plane. Therefore, the authors attempted to refine the CNM as well. The authors in [22] put efforts into controlling the rate without a feedback channel. Based on the motion intensity, the algorithm adjusted the rates by switching the Laplacian distribution parameters in CNM between frame and block levels. The authors in [23] proposed an adaptive low computational DVC, which estimated correlation by using the expectation propagation during the channel decoding process. The correlation estimation is carried out jointly with decoding of the factor graph during the channel decoding process.

In [24], the authors proposed a CNM that is independently executed at both the encoder and decoder at no cost of extra computation at the encoder. At the encoder, the CNM calculates the number of least significant bits that are required to be sent to the decoder, assuming that the remainder is determined at the decoder. In [25], the authors proposed a hash-based DVC where the correlation noise is statistically dependent on the SI. The proposed algorithm performs online estimation of the SI-dependent correlation noise parameter at the transform coefficient bands level. Additionally, the algorithm successively refines the correlation parameter after decoding every bit-plane.

The authors in [26] attempted to reduce the deviation between the Laplacian statistical distribution model and the small and large residual coefficients by proposing a hybrid distribution correlation noise model. This hybrid model is based on the K-Medoids and Cauchy distributions clustering, which clusters all residual coefficients into small and large coefficients clusters. The small coefficients cluster is K-Medoids modeled to

improve Laplacian distribution, while the large residual coefficients are modeled with Cauchy distribution.

On the other hand, the codec in [27] used the parallel LDPC decoding to estimate the correlation noise parameters and decode on the factor graph. This algorithm estimated one parameter for each band. The codec in [11] extended the work of [27] and proposed a recursive variational Bayes factor graph. Furthermore, it deployed a new message-passing algorithm to decode bit-planes corresponding to each band and simultaneously estimates and refines the correlation noise parameter.

The correlation noise parameter estimation plays an important role in achieving coding efficiency. In most codecs, the highly computational motion-compensated interpolation (MCTI) [10] is performed for the SI generation, and the resultant motion vectors are utilized for the correlation noise parameter estimation. However, in the codec presented in [15], an empirical study of a phase-based fast frame interpolation algorithm was conducted to effectively generate a low computational and fast SI in DVC when motion vectors are not available. For such codecs, there is a need to design an online CNM framework so it leads to the coding efficient codec framework along with low computational SI-generation feature. In this paper, our main focus is to design an online CNM framework for such codecs that will lead to the development of a full DVC codec with less decoding computational complexity and can compete with DIS in terms of coding efficiency and RD performance. Therefore, in this study, an attempt has been made to design a framework for online estimation of CNM for the Laplacian distribution parameter while considering the mean of the coefficients band.

### 3. Proposed DVC Framework for Online Correlation Noise Model

This section addresses the basic concept and implementation of the proposed codec and presents the proposed online correlation noise model methodology. We will first present the general concept of the proposed online correlation noise model based on the mean of band coefficients before moving on to the mathematical modeling of the proposed model.

#### 3.1. General Concept of Proposed Online Correlation Noise Model

Figure 1 depicts the proposed DVC framework for calculating the online correlation noise model CNM for the DVC frameworks [10,15]. In [15], the motion estimation is not exploited at the decoder, and the actual frame is also not available at the decoder. Therefore, online CNM computation is quite a challenging task in such a DVC framework. Keeping that in mind, we proposed a novel online CNM based on the mean of band coefficients. In this framework, the Band Mean Calculator (BMC) is deployed at the encoder and decoder to perform the necessary calculations which are used in the CNM.

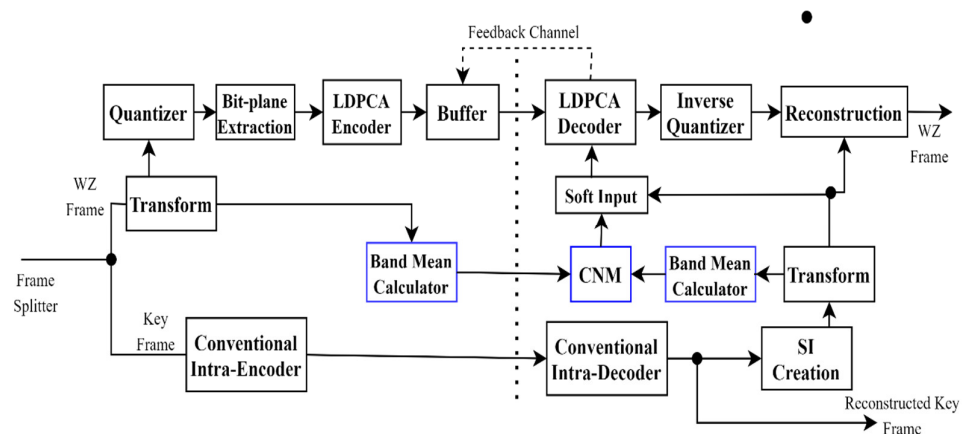


Figure 1. Proposed DVC framework with novel online correlation noise model.

At the encoder, the BMC calculates the mean of the coefficients of each band of the WZF and sends them to the decoder. The BMC deployed at the decoder calculates the mean of each coefficient band of side information (SI) in a similar way as it does on the encoder. The Laplacian distribution is used to model the residual between the means of the original WZ and the corresponding SI frame for each band. The CNM calculation details and mathematical representation are given in Section 3.2. The SI generation process from [15] and conventional interpolation in [10] were used in this study. A brief overview of the phase-based SI generation algorithm from [15] is presented in Algorithm 1. We deployed and verified the performance of this novel online CNM concept for the DIS codec and named it *Distributed Video Coding with Online Band Mean Correlation Noise Model* (DIVCOM). Next, the same online CNM concept is deployed for the DVC codec with the phase-based SI [15], and we called the implementation *Phase-based Distributed Video Coding with Online Band Mean Correlation Noise Model* (PDIVCOM).

### 3.2. Mathematical Representation of Phase Interpolation for Side Information Generation

The standard interpolation approaches, e.g., optical flow, require accurate pixel correspondences between images to interpolate the in-between frames. In DVC, motion-compensated temporal interpolation (MCTI) is usually used for SI generation. However, in this paper, we implemented the phase-based interpolation for SI at the decoder. This technique represents motion as phase shifts in individual pixels and the computational cost is a fraction of that of any of the traditional SI interpolation techniques. In particular, the phase-based technique decomposes the frames into local phase and amplitude parameters using the complex-valued steerable pyramid. This complex steerable pyramid decomposes the one-dimensional (1D) or two-dimensional (2D) signal into spatial scale, orientation, and position [28]. In videos, the phase-based interpolation method manipulates motion by analyzing the signal of the local phase over time in different spatial scales and orientations. It mitigates the need for highly complex global computation. The computation efficiency is offered by phase-based representation, as it only performs per-pixel modifications to represent the pixel motion by shifting its phase [29]. Further efficiency is achieved due to the phase-shift correction method that combines the phase information across different levels of a multi-scale pyramid. The successfulness of image transition is achieved due to the deployment of a correction algorithm which adapts both amplitude and phase shift of the image. Before starting the part of actual image interpolation, we first explain its working concept for a basic one-dimensional (1D) signal.

The Fourier shift theorem motivates the assumption that motion can be encoded using the phase differences. Let us consider the 1D case, in which any function  $f(x)$  can be represented in the Fourier domain as a sum of complex sinusoids over all frequencies  $\omega$  by Equation (1),

$$f(x) = \sum_{\omega=-\infty}^{\omega=+\infty} A_{\omega} e^{i\omega x} = \sum_{\omega=-\infty}^{\omega=+\infty} A_{\omega} e^{i\varnothing_{\omega}} \quad (1)$$

where the  $\varnothing_{\omega}$  and  $A_{\omega}$  represent the phase and amplitude of the complex sinusoid, respectively. Then the shifted version of function  $f(x)$  shifted by spatial displacement  $\delta(t)$  of is given by Equation (2).

$$f(x + \delta(t)) = \sum_{\omega=-\infty}^{\omega=+\infty} A_{\omega} e^{i\omega(x+\delta(t))} \quad (2)$$

The phase difference,  $\varnothing_{diff}^{\omega}$ , between the original and shifted functions can be represented by Equation (3).

$$\varnothing_{diff}^{\omega} = \omega x - \omega(x - \delta(t)) = \omega \delta(t) \quad (3)$$

The phase shift  $\mathcal{O}_{shift}$ , which corresponds to the actual spatial displacement between translated functions is defined in terms of the phase difference  $\mathcal{O}_{diff}^\omega$  between two phase curves scaled by angular frequency  $\omega$  and is given in Equation (4).

$$\mathcal{O}_{shift} = \frac{\mathcal{O}_{diff}^\omega}{\omega} \tag{4}$$

In the Fourier transform domain, the shifted function is defined as the sum of complex sinusoids over all frequencies  $\omega$ , and is given by Equation (5),

$$f(x + \delta(t)) = \sum_{\omega=-\infty}^{\omega=+\infty} R_\omega(x, t) \tag{5}$$

where each sinusoid represents one band, i.e.,  $R_\omega(x, t) = A_\omega e^{i\omega(x+\delta(t))}$ . Now for the intermediate sinusoids representing the translational functions, the phase difference is modified according to the intermediate positions between functions that are defined by weight  $\beta \in (0, 1)$ . Therefore, the modified bands  $\check{R}_\omega(x, t)$  with corresponding modified phase  $\mathcal{O}_\omega = \omega(x + \beta\delta(t))$  w.r.t  $\beta$  is given by Equation (6).

$$\check{R}_\omega(x, t) = A_\omega e^{i\omega(x+\beta\delta(t))} \tag{6}$$

The in-between functions are then taken by integrating the modified bands following Equation (5).

### 3.2.1. Two Dimensional Functions—General Mathematical Representation

In two-dimensional (2D) functions, the sinusoids are separated into bands according to frequency  $\omega$ , as well as the spatial phase orientations i.e., the even and odd phase orientations, by using the complex-valued steerable pyramid filter [30–32]. This filter decomposes the input images into several oriented frequency bands  $R_{\omega,\theta}$ , and thus allows for meaningful phase measurement from coefficients of a pyramid. The real part of each coefficient is represented in cosine form and known as even-symmetric filtered, whereas the imaginary part is in sine filter representation, known as an odd-symmetric filtered. The steerable pyramids that are non-oriented and not captured during the pyramid levels are summarized as real-valued high and low-pass residual signal components.

#### 3.2.1.1. Phase Computation Steps

During the phase computation, the complex-valued response  $R_{\omega,\theta}$  obtained by applying steerable filters  $\psi_{\omega,\theta}$  [30] on image  $I$ , and are represented in Equations (7)–(9),

$$R_{\omega,\theta}(x, y) = (I * \psi_{\omega,\theta})(x, y) \tag{7}$$

$$= A_{\omega,\theta}(x, y) e^{i\mathcal{O}_{\omega,\theta}(x,y)} \tag{8}$$

$$= C_{\omega,\theta}(x, y) + i S_{\omega,\theta}(x, y) , \tag{9}$$

where  $C_{\omega,\theta}$  and  $S_{\omega,\theta}$  define the cosine and sine parts, respectively. As stated before, the cosine part,  $C_{\omega,\theta}$  and the sine part,  $S_{\omega,\theta}$  represent even- and odd-symmetric filter responses, respectively. Therefore, the amplitude and phase components can be computed by Equations (10) and (11), respectively.

$$A_{\omega,\theta}(x, y) = \sqrt{C_{\omega,\theta}(x, y)^2 + S_{\omega,\theta}(x, y)^2} \tag{10}$$

$$\mathcal{O}_{\omega,\theta}(x, y) = \arctan\left(\frac{S_{\omega,\theta}(x, y)}{C_{\omega,\theta}(x, y)}\right) \tag{11}$$

### 3.2.1.2. Phase Difference Calculation Steps

As stated earlier, in phase interpolation, the motion is represented by a phase shift; therefore, interpolating the phase shift requires the phase difference  $\varnothing_{diff}$  between the phases of two input images,  $\varnothing_1$  and  $\varnothing_2$  and is computed by Equation (12),

$$\varnothing_{diff} = \text{atan2}(\sin(\varnothing_1 - \varnothing_2), \cos(\varnothing_1 - \varnothing_2)), \tag{12}$$

where  $\text{atan2}(\cdot)$  represents the four-quadrant inverse tangent and leads to angular values between  $[-\pi, \pi]$ , which corresponds to the angular difference between two input image phases. Additionally, it determines the limit of motion, which is bounded by Equation (13),

$$|\varnothing_{shift}| = \frac{|\varnothing_{diff}|}{\omega} \leq \frac{\pi}{\omega} \tag{13}$$

where  $\omega = 2\pi v$ , and  $v$  denotes the spatial frequency of the pyramid level. In a multi-scale pyramid, each level represents a particular band of spatial frequencies  $v \in [v_{min}, v_{max}]$ . Let  $v_{max}$  correspond to the highest frequency of any level then a phase difference of  $\pi$  represents a shift of one pixel. This appears to be a reasonable shift in coarser pyramid levels for low-frequency content; however, for high-frequency content, it is too limiting to achieve realistic interpolation.

### 3.2.1.3. Phase Shifting and Correction

To avoid the phase ambiguity which happens due to large displacement corresponding to a phase difference of more than  $\pi$ , and since the phase is periodic, the phase difference is normalized to between  $[-\pi, \pi]$ .

Interpolation works accurately as long as the shift computed on a particular level mainly captures the frequency content corresponding to the correct motion. Therefore, shift correction based on confidence estimation is deployed to correctly estimate the motion even for high-frequency content. This approach robustly interpolates the accurate motion for high frequency by taking all available shift information into account. This approach assumes that phase difference between two resolution levels does not differ arbitrarily, i.e., phase differences between levels can be used as a confidence measure that quantifies whether the estimated phase shift is admissible.

The phase shift correction is performed on the level  $l$  if the computed shift at level  $l$  differs from the coarser level  $l + 1$  by more than a threshold. The phase shift correction is performed by first adding multiples of  $\pm 2\pi$  to  $\varnothing_{diff}$ . This leads to absolute differences between the phase values of consecutive levels that can never be greater than a given tolerance. The  $\pi$  is used as tolerance distance which modifies the phase values in such a way that the phase difference of a pixel between two levels can never be larger than  $\pi$ . This step allows a meaningful range extension because the original phase difference are truncated to the range  $[-\pi, \pi]$ . The actual shift correction depends on the difference between two levels, which is taken under consideration for confidence estimation and is calculated using Equation (14),

$$\varphi = \text{atan2}\left(\sin\left(\varnothing_{diff}^l - \varnothing_{diff}^{l+1}\right), \cos\left(\varnothing_{diff}^l - \varnothing_{diff}^{l+1}\right)\right), \tag{14}$$

where the coarser level phase value is scaled according to the arbitrary pyramid scale factor  $\lambda > 1$ . When  $|\varphi| > \frac{\pi}{2}$ , the shift correction is performed to obtain the corrected phase difference using (15) and considerably leads to a better interpolation result.

$$\tilde{\varnothing}_{diff}^l = \lambda \varnothing_{diff}^{l+1} \tag{15}$$

Although the phase shift correction can model large motion, it still has the limitation of presenting blur artifacts in some motions. Therefore, an additional enhancement step is

performed to limit even admissible phase shifts into well representable motions. Therefore, the phase difference is limited by a constant,  $\emptyset_{limit}$ , defined by Equation (16)

$$\emptyset_{limit} = \tau\pi\lambda^{L-l} \quad (16)$$

where  $\tau$  defines the percentage of limitation and  $\tau \in (0, 1)$ ,  $L$  is the total number of levels,  $\lambda$  is the scale factor and  $l$  is the current level. At the coarser level, if the magnitude of phase difference exceeds  $\emptyset_{limit}$  then the corrected phase difference is set to zero.

The next step is a smooth-phase interpolation between phases of input image 1,  $\emptyset_1$ , and input image 2,  $\emptyset_2$ . For convenience, from here onwards we will drop the superscript  $l$  to represent the pyramid level. As a result of shift correction, there is no surety that  $\emptyset_1 + \emptyset_{diff}$  matches  $\emptyset_2$  or any of the equivalent multiples of  $\emptyset_2 + \gamma 2\pi$  where  $\gamma \in \mathbb{N}_0$ . For smooth interpolation, the original phases  $\emptyset_1$  and  $\emptyset_2$  are preserved along with shift-corrected phase difference  $\tilde{\emptyset}_{diff}$ . Therefore, phase adjustment is done using

$$\check{\emptyset}_{diff} = \emptyset_{diff} + \gamma 2\pi \quad (17)$$

to give us the adjusted phase difference,  $\check{\emptyset}_{diff}$ , where the  $\gamma$  is calculated using

$$\gamma = \underset{\gamma}{\operatorname{argmin}} \left\{ \left( \tilde{\emptyset}_{diff} - \left( \emptyset_{diff} + \gamma 2\pi \right) \right)^2 \right\}. \quad (18)$$

After the phase adjustment, the phase for the interpolated image,  $\emptyset_\beta$ , is computed using Equation (19) by taking adjusted phase difference  $\check{\emptyset}_{diff}$  and the phase of one of the images, e.g.,  $\emptyset_1$ .

$$\emptyset_\beta = \emptyset_1 + \beta \check{\emptyset}_{diff} \quad (19)$$

The next step is a reconstruction of the interpolated image which requires the interpolated phase and the interpolated amplitude are required for smooth interpolation [33].

Algorithm 1 summarizes the execution steps involved in the interpolation of SI using the phase interpolation method. The inputs given to start the execution process are the previous and next keyframes (KFs) and the interpolation parameter,  $\beta$ . The previous and next KFs are represented by  $I_1$  and  $I_2$  respectively. The interpolated SI is denoted by  $I_{SI}$ . The phase interpolation process is initialized with steerable pyramid decompositions of both keyframes ( $I_1$  and  $I_2$ ) and calculation of their respective amplitudes ( $A_1$  and  $A_2$ ). Later, the corresponding phases,  $\emptyset_1$  and  $\emptyset_2$ , and phase differences are calculated. For smooth interpolation of  $I_{SI}$ , the level-by-level shift correction and phase adjustment of the phase difference is performed. The phase of the interpolated image,  $\emptyset_\beta$  and its amplitude,  $A_\beta$ , are interpolated in steps 7 and 8, respectively, then recombined to generate the respective pyramids. Finally, all interpolated pyramids are utilized to reconstruct the interpolated image,  $I_{SI}$ .

### 3.2.2. Mathematical Representation of Proposed Online Correlation Noise Model Estimation

In conventional video coding, the distribution of the motion-compensated residual coefficients is modeled using the Laplacian distribution [34]. Different distribution models such as Gaussian distribution are found in the literature; however, the Laplacian distribution is often chosen to obtain a good trade-off between model complexity and accuracy [35]. Due to this trade-off feature, the Laplacian distribution is widely adopted in DVC for modeling the CNM.



---

**Algorithm 1** Side Information Generation with Phase Interpolation Process

---

**INPUTS.** Two input images are:  $I_1$  and  $I_2$  represent the previous and next keyframes (KFs), respectively.

Interpolation parameter:  $\beta$

**INITIALIZATION.** Steerable pyramid decompositions:  $P_1$  and  $P_2$

Amplitudes Calculation:  $A_1$  and  $A_2$

**OUTPUT.** Output (interpolated Side Information):  $I_{SI}$

**Step 1.**  $(P_1, P_2) \leftarrow$  Decompose  $(I_1, I_2)$  Equations (7)–(9), Refer [30]

**Step 2.**  $(A_1, A_2) \leftarrow$  Amplitude  $(I_1, I_2)$  Equation (10)

**Step 3.**  $(\emptyset_1, \emptyset_2) \leftarrow$  Phase  $(P_1, P_2)$  Equation (11)

**Step 4.**  $\emptyset_{diff} \leftarrow$  Phase Difference  $(\emptyset_1, \emptyset_2)$  Equation (12)

**Step 5.** **for all**  $l = L - 1 : 1$  **do**

$\tilde{\emptyset}_{diff}^l \leftarrow$  Shift Correction  $(\tilde{\emptyset}_{diff}^{l+1})$  Section 3.2.1.3

**end for**

**Step 6.**  $\check{\emptyset}_{diff} \leftarrow$  Adjust Phase  $(\emptyset_{diff}, \tilde{\emptyset}_{diff})$  Equation (17)

**Step 7.**  $\emptyset_\beta \leftarrow$  Interpolate  $(\emptyset_1, \check{\emptyset}_{diff}, \beta)$  Equation (19)

**Step 8.**  $A_\beta \leftarrow$  Blend  $(A_1, A_2, \beta)$  Section 3.2.1.3, Refer [33]

**Step 9.**  $P_\beta \leftarrow$  Recombine  $(\emptyset_\beta, A_\beta)$  Refer [30]

**Step 10.**  $I_{SI} \leftarrow$  Reconstruct  $(P_\beta)$

---

Ideally, the calculation of the noise correlation parameter can either be done with the SI at the encoder or the actual WZF at the decoder. However, this is practically not possible, so for the online correlation noise parameter estimation, the different codecs exploited the temporal correlation at the decoder [18] at the cost of extra computation. In this work, we propose a novel frame-level online CNM parameter calculation method using transformed coefficient bands. For offline CNM, the Laplacian distribution of the residual of WZF and SI transformed coefficient at  $(x, y)$ , with Laplacian distribution parameter,  $\alpha$ , is found using (20).

$$p[WZ(x, y) - SI(x, y)] = \frac{\alpha}{2} \exp[-\alpha |WZ(x, y) - SI(x, y)|] \quad (20)$$

where  $\alpha = \sqrt{\frac{2}{\sigma^2}}$  is used to determines the error distribution between  $WZ(x, y)$  and  $SI(x, y)$ , where  $\sigma^2$  defines the variance [19].

To compute the online CNM using the mean of transformed coefficients of a band at the decoder, it is necessary to estimate the Laplacian distribution parameter which is necessary to determine the error distribution between the original frame and SI. Therefore, for the proposed frame-level Laplacian distribution model, the estimation of the Laplacian distribution parameter is a crucial step. Since, for our proposed method, the Laplacian distribution model is computed at the frame level, the Laplacian distribution parameter is denoted as  $\alpha_f$ . The steps involved in estimating  $\alpha_f$  are elaborated next.

Let  $\mu^R$  be the set of means of residuals of the coefficient bands,  $\mu_b^R$ . Therefore, for a total of 16 coefficient bands in a frame,

$$\{ \mu_b^R | \mu_b^R \in \mu^R, b = 1, 2, \dots, 16 \} \quad (21)$$

and

$$\mu_b^R = \mu_b^{WZ} - \mu_b^{SI}, b = 1, 2, \dots, 16, \quad (22)$$

where  $\mu_b^{WZ}$  and  $\mu_b^{SI}$  are the means of the WZF and SI transformed coefficient bands, respectively. The  $\mu_b^{WZ}$ , is calculated at the encoder using

$$\mu_b^{WZ} = \frac{1}{M} \sum_{j=1}^M X_{DCT,b}(j). \quad (23)$$

In (23),  $X_{DCT,b}(j)$  is the transformed coefficient of the WZF for the  $j$ -th coefficient in band  $b$ . On the other hand, the value,  $\mu_b^{SI}$ , is the mean of transformed coefficients of SI band and is calculated at the decoder by using

$$\mu_b^{SI} = \frac{1}{M} \sum_{j=1}^M Y_{DCT,b}(j). \tag{24}$$

In (24),  $Y_{DCT,b}(j)$  is the transformed coefficient of the SI for the  $j$ -th coefficient in band  $b$ . In both (23) and (24),  $j = 1, 2, \dots, M$  where  $M$  defines the total number of coefficients in a band.

Next, the variance,  $\sigma_f^2$ , for the proposed residual means,  $\mu^R$ , is calculated using the equation

$$\sigma_f^2 = E\left[(\mu^R)^2\right] - \left(E\left[\mu^R\right]\right)^2, \tag{25}$$

where  $E[\mu^R]$  and  $E[(\mu^R)^2]$  are computed using Equations (26) and (27), respectively.

$$E\left[\mu^R\right] = \sum_{b=1}^{16} \mu_b^R P_b \tag{26}$$

$$E\left[(\mu^R)^2\right] = \sum_{b=1}^{16} (\mu_b^R)^2 P_b \tag{27}$$

In (26) and (27),  $P_b$  denotes the probability of occurrence of  $\mu_b^R$  in  $\mu^R$ . Finally, the Laplacian distribution parameter  $\alpha_f$  is calculated using

$$\alpha_f = \begin{cases} \frac{1}{2 \times 10^2 \times (\sigma_f^2)}, & 0 < \sigma_f^2 \leq 1 \\ \frac{1}{10 \times \mu^{Rsum}}, & 1 < \sigma_f^2 < 100 \ \& \ \mu^{Rsum} \leq 50 \\ \frac{1}{20 \sqrt{\mu^{Rsum}}}, & 1 < \sigma_f^2 < 100 \ \& \ \mu^{Rsum} > 50 \\ \frac{1}{\mu^{Rsum}}, & \sigma_f^2 \geq 100 \end{cases} \tag{28}$$

where  $\mu^{Rsum}$  is the sum of absolute values of all residuals in  $\mu^R$  and given as

$$\mu^{Rsum} = \sum_{b=1}^{16} \left| \mu_b^R \right|. \tag{29}$$

In (29),  $\left| \mu_b^R \right|$  represents the absolute mean value of the residual between the WZF and its corresponding SI for coefficient band  $b$ , where  $b = 1, 2, \dots, 16$ . Therefore, for the frame-level Laplacian distribution, Equation (20) becomes

$$p\left[\left| \mu_b^R \right| \right] = \frac{\alpha_f}{2} \exp\left[-\alpha_f \left| \mu_b^R \right| \right]. \tag{30}$$

Algorithm 2 is used to perform the step-by-step calculation of the Laplacian distribution parameter  $\alpha_f$  under the proposed CNM methodology. The process starts by computing the values of  $\mu_b^{WZ}$ ,  $\mu_b^{SI}$  and subsequently,  $\mu_b^R$  for all  $b$ . All the values of  $\mu_b^R$  are then used in the computation of the variance,  $\sigma_f^2$ , in step 4. Next, the Laplacian distribution parameter  $\alpha_f$  is calculated in step 5. Finally, the procedure is repeated for subsequent frames.

**Algorithm 2** Determine the Steps for Calculation of Laplacian Distribution Parameter  $\alpha_f$  under Proposed Online Correlation Noise Model Methodology

**INPUTS.** Transformed Coefficients of Band-  $X_{DCT}(j)$ ,  $Y_{DCT}(j)$

**INITIALIZATION.** Mean of coefficient bands of WZF

**OUTPUTS.**  $\alpha_f$

Step 1. **for all**  $b = 1 : 16$  **do**  
     Calculate  $\mu_b^{WZ}$  Equation (23)  
**end for**

Step 2. **for all**  $b = 1 : 16$  **do**  
     Calculate  $\mu_b^{SI}$  Equation (24)  
**end for**

Step 3. Calculate Residual of corresponding  $\mu_b^{WZ}$  and  $\mu_b^{SI}$   
     **for all**  $b = 1 : 16$  **do**  
          $\mu_b^R \leftarrow \mu_b^{WZ} - \mu_b^{SI}$  Equation (22)  
     **end for**

Step 4. Calculate  $\sigma_f^2$   
      $\sigma_f^2 \leftarrow E[(\mu^R)^2] - (E[\mu^R])^2$  Equations (25)–(27)

Step 5. Calculate  $\alpha_f$  Equation (28)  
      $\mu^{Rsum} \leftarrow \sum_{b=1}^{16} |\mu_b^R| \quad \forall b$  Equation (29)  
     **if**  $0 < \sigma_f^2 \leq 1$  **then**  
          $\alpha_f \leftarrow \frac{1}{2 \times 10^2 \times (\sigma_f^2)}$   
     **else if**  $1 < \sigma_f^2 < 100$  and  $\mu^{Rsum} \leq 50$  **then**  
          $\alpha_f \leftarrow \frac{1}{10 \times \mu^{Rsum}}$   
     **else if**  $1 < \sigma_f^2 < 100$  and  $\mu^{Rsum} > 50$  **then**  
          $\alpha_f \leftarrow \frac{1}{20 \sqrt{\mu^{Rsum}}}$   
     **else**  $\sigma_f^2 > 100$  and  $\mu^{Rsum} \sqrt{R}$  **then**  
          $\alpha_f \leftarrow \frac{1}{\mu^{Rsum}}$   
     **end if**

Step 6. **REPEAT STEPS 1–5 FOR THE UPCOMING FRAME**

## 4. Results and Discussion

### 4.1. Experimental Setup

The experiments are carried out on a system with Core (TM) i7-7820HQ, 64-bit OS, CPU 2.90 GHz, and RAM 32 GB specifications. The results are compiled for the different test video sequences. The performance is analyzed by running the experiments on full video sequences with a frame rate (fps) of 15 Hz and a group of pictures (GOP) of size 2. The experiments were performed using six different values of the quantization metric,  $Q_m$ , for quantizing the WZF to achieve different output qualities. The definitions and values of  $Q_m$  used in the proposed codecs are the same as the ones used by DIS in [10]. A list of abbreviations for the different codecs evaluated is given in Table 1. Different parameters are involved in phase-based interpolation for SI generation. Parameter setting is required to obtain the best SI quality. Therefore, the following parameter settings are adopted for SI generation; the number of pyramid levels is 17, the phase shift is 0.4 radians, the number of orientations is 12, and the pyramid scale is  $(0.4)^{\frac{1}{4}}$ .

**Table 1.** Abbreviations of different video codecs.

Abbreviation	Description
DIS	DISCOVER Codec
DIVCOM	Distributed Video Coding with Online Band Mean Correlation Noise Model
PDIVCOM	Phase-based Distributed Video Coding with Online Band Mean Correlation Noise Model

#### 4.2. Performance Evaluation

Four video sequences, i.e., “Coastguard”, “Akiyo”, “Foreman” and “Hall”, were used to compare the performance of the proposed codecs, against the DIS and Intra H.264 codecs. For each video sequence, the average PSNRs and coding rates are determined for different values of the quantization metric,  $Q_m$ . The average PSNR and coding rate of Intra H.264, DIS, DIVCOM, and PDIVCOM were computed for a frame rate of 15 Hz. Next, the performance of the Wyner–Ziv coder is analyzed by evaluating the average channel-decoding rate required to correct the errors of SI for successfully decoding the WZF. Finally, the rate distortion (RD) performance is compared between Intra H.264, DIS, DIVCOM, and PDIVCOM.

##### 4.2.1. “Coastguard” Video Sequence

From Table 2, we can observe that in most cases, DIVCOM gives a better average PSNR than DIS, while PDIVCOM gives a better average PSNR than DIS in all cases. PDIVCOM outperforms DIVCOM for lower values of  $Q_m$ , while DIVCOM outperforms PDIVCOM at higher values of  $Q_m$ . In terms of coding efficiency, the coding rate of DIVCOM is better than DIS by between 0.687 kbps to 8.05 kbps, while the coding rate of PDIVCOM is better than DIS by between 1.49 kbps to 8.955 kbps. However, closer analysis at the frame-to-frame level reveals that the RD performance of the proposed codecs is reduced in some frames of this video test sequence. Even so, the decrease in PSNR is small when taking into consideration the improvement in the coding rate.

**Table 2.** Visual quality and rate performance evaluation of different codecs for “Coastguard” video sequence at frame rate of 15 Hz.

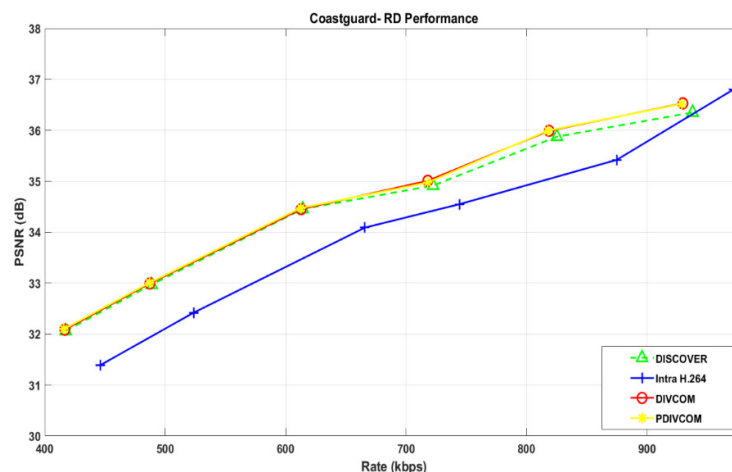
Video Sequence	Quantization Matrix, $Q_m$	Average PSNR (dB)			Coding Rate (kbps)		
		DIS	DIVCOM	PDIVCOM	DIS	DIVCOM	PDIVCOM
Coastguard	1	32.0616	32.0867	32.0978	417.431	416.744	415.934
	2	32.97	32.9945	33.008	488.7012	487.3013	487.1401
	3	34.455	34.4520	34.4740	614.1899	612.8306	612.5757
	4	34.9146	35.009	34.974	722.3970	718	717.7109
	5	35.8749	35.9899	35.9942	825.1416	818.6396	817.7002
	6	36.35	36.5285	36.5175	937.7612	929.7104	928.8057

Table 3 presents the results for the Wyner–Ziv (WZ) coder part for decoding WZF. For DIVCOM, the average PSNR gain per WZF over DIS ranges from 0.0521 dB to 0.36 dB, while for PDIVCOM, the gain ranges from 0.039 dB to 0.3323 dB. The average channel-decoding rate results indicate that DIVCOM achieves an average coding efficiency value between 101 bits to 1.151 kbits, whereas the average coding efficiency of PDIVCOM ranges from 219 bits to 1.28 kbits. At the individual frame level, the RD performance of the proposed codecs is degraded in some frames. However, overall, these codecs perform well for most of the frames in the test sequence. From Table 3, we can see that DIVCOM gave a negative average PSNR gain for quantization point 3. The reduction in performance is due to the CNM parameter in the reconstruction process. Even so, the average decoding rate indicates that better coding efficiency is achieved by DIVCOM. For the higher quantization points, the average coding rates of DIVCOM and PDIVCOM are close to that of DIS because, at these higher points, a variant coding rate behavior is noticed for the proposed codecs. For some frames of the sequence, a bit rate per frame that is close to or higher than DIS is achieved; however, higher coding efficiency is attained for most of the frames.

**Table 3.** Coding efficiency evaluation of proposed novel online CNM for “Coastguard” video sequence.

Video Sequence	$Q_m$	Average Channel Decoding Rate per WZF (bits)			Average PSNR Gain per WZF Compared to DIS (dB)	
		DIS	DIVCOM	PDIVCOM	DIVCOM	PDIVCOM
Coastguard	1	4545	4444	4326	0.0521	0.0751
	2	5404	5199	5176	0.0731	0.0449
	3	7722	7486	7523	−0.006	0.0396
	4	18,070	17,425	17,384	0.1957	0.1231
	5	22,979	22,028	21,891	0.2417	0.2507
	6	33,856	32,678	32,546	0.3549	0.3323

Figure 2 depicts the experimental results in the form of RD performance carried out with Intra H.264, DIS, DIVCOM, and PDIVCOM on the “Coastguard” video. It can be seen that for all quantization points, the performance of DIVCOM and PDIVCOM is comparable to DIS. For low quantization points, the DIVCOM gives a coding efficiency of roughly between 1 and 2.5 kbps along with a significant improvement in the PSNR. The PDIVCOM also shows almost the same performance trend in both coding efficiency and PSNR. At higher quantization points, both DIVCOM and PDIVCOM outperform DIS. DIVCOM achieves a coding efficiency of up to 8 kbps and an improvement in PSNR of 0.18 dB over DIS. PDIVCOM performs even better than DIVCOM and achieves a coding efficiency of up to 9 kbps and PSNR gain of up to 0.17 dB. The small improvement in coding efficiency at lower quantization points is due to fewer bands undergoing the decoding process. However, the improvement becomes significant at higher quantization points because more bands undergo the decoding process. Upon further evaluation of the frames for each quantization point, a higher coding rate is observed for a few of the frames in the test sequence. This is due to a miscalculation of the CNM parameter by the proposed model. Even the PSNR quality is also compromised for those frames. The performance comparison of DIVCOM with Intra H.264 shows that DIVCOM achieves better coding efficiency for all quantization points with a bit rate saving of up to 41.92 kbps. For low to intermediate quantization points, DIVCOM achieves a PSNR gain of up to 0.69 dB. However, at the highest quantization point, the PSNR degrades by up to 0.28 dB. PDIVCOM shows better coding efficiency for all quantization points compared to Intra H.264 with bit rate savings ranging from 30.23 kbps to 42.83 kbps. At the same time, PDIVCOM achieves PSNR gains up to 0.71 dB for all quantization points. However, similar to DIVCOM its PSNR performance degrades at the highest quantization point by 0.29 dB.

**Figure 2.** Rate-distortion (RD) performance graphical evaluation of “Coastguard” video sequence with frame rate of 15 Hz and GOP 2.

#### 4.2.2. “Akiyo” Video Sequence

Table 4 presents the experimental results of the “Akiyo” video sequence. From the table, it can be seen that the improvement in PSNR of DIVCOM over the DIS codec is between 0.007 dB and 0.021 dB, while PDIVCOM gives an improvement in PSNR between 0.037 dB and 0.12 dB over the DIS codec. Even though the coding efficiency of DIVCOM improves by between 0.38 kbps and 1.7 kbps over DIS codec. In some cases, when the variance is less than 0.1, the coding rate of DIVCOM is worse than DIS. The coding efficiency of PDIVCOM is better than DIS by between 1.53 kbps and 10.938 kbps. On a frame-by-frame base analysis, PDIVCOM is better than DIVCOM because it has fewer occurrences of frames that have worse PSNR and coding rates than DIS. Overall, the PDIVCOM outperforms other codecs investigated for this test video sequence.

**Table 4.** Visual quality and rate performance evaluation of different codecs for “Akiyo” video sequence at frame rate of 15 Hz.

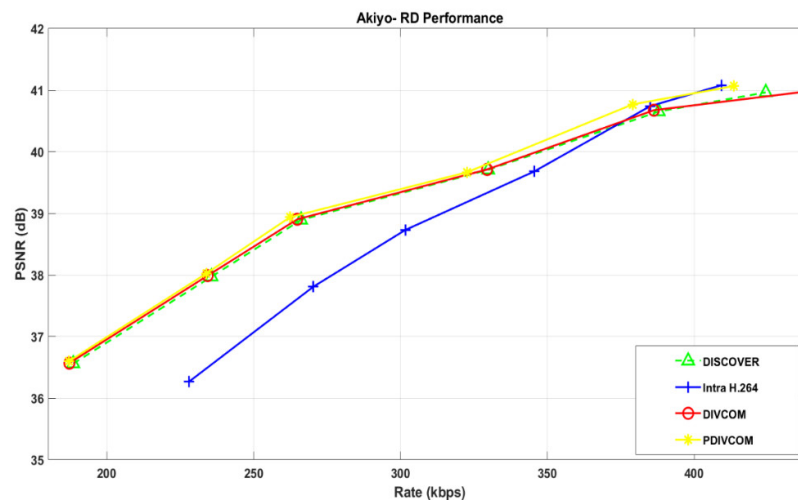
Video Sequence	$Q_m$	Average PSNR (dB)			Coding Rate (kbps)		
		DIS	DIVCOM	PDIVCOM	DIS	DIVCOM	PDIVCOM
Akiyo	1	36.5622	36.5688	36.5988	188.5562	187.2031	187.0215
	2	37.9728	37.9896	38.0227	235.6357	234.3213	233.8198
	3	38.8950	38.9045	38.9394	266.1543	264.8397	262.4297
	4	39.7082	39.7128	39.6677	329.874	329.4976	322.6880
	5	40.6535	40.6747	40.7688	387.9673	386.311	379.0576
	6	40.9639	40.9827	41.0656	424.4033	439.0259	413.4653

Table 5 presents the results for the Wyner–Ziv (WZ) coder part for decoding WZF. For DIVCOM, the average PSNR gain per WZF over DIS ranges from 0.0137 dB to 0.413 dB, while for PDIVCOM, the gain ranges from 0.0654 dB to 0.24 dB. The average channel-decoding rate results indicate that DIVCOM achieves an average coding efficiency value between 55 bits and 243 kbits. However, in one case where a few frames had a variance of less than 0.1, additional bits were requested several times to mitigate the errors. This resulted in DIVCOM having a worse coding rate than DIS. PDIVCOM achieves an average coding rate between 219 bits and 1.28 kbits. It can also be observed that for this video sequence PDIVCOM outperforms both DIS and DIVCOM in terms of coding efficiency. The gain comparison shows that there are only slight PSNR gains acquired by both proposed codecs. This is because the estimated CNM parameter used in the decoding process is not suitable for some of the frames with a variance of around 0.1, so this results in inaccurately reconstructed frames.

**Table 5.** Coding efficiency evaluation of proposed novel online CNM for “Akiyo” video sequence.

Video Sequence	$Q_m$	Average Channel Decoding Rate per WZF (bits)			Average PSNR Gain per WZF Compared to DIS (dB)	
		DIS	DIVCOM	PDIVCOM	DIVCOM	PDIVCOM
Akiyo	1	2733	2535	2509	0.0137	0.0654
	2	3436	3244	3170	0.035	0.1034
	3	4309	4116	3765	0.0197	0.0921
	4	8674	8619	7623	0.0094	−0.084
	5	10,417	10,174	9113	0.044	0.2389
	6	13,732	15,871	12,132	0.0413	0.2097

The graph in Figure 3 illustrates the RD performance of the codecs on the “Akiyo” test video sequence at all quantization points. Both DIVCOM and PDIVCOM perform better than DIS by having a better coding efficiency and PSNR at all quantization points. From the RD graph, it is noticed that for lower quantization points, the DIVCOM achieves a coding efficiency of up to 1.31 kbps and PSNR gain of up to 0.017 dB. However, the DIVCOM coding efficiency slightly reduces at higher quantization points but still manages to achieve a PSNR gain of 0.021 dB for these points. PDIVCOM achieves a higher coding efficiency than both DIS and DIVCOM. At low quantization points, the coding efficiency of PDIVCOM can go up to 3.71 kbps, and up to 11 kbps at higher quantization points. The PSNR gain achieved by PDIVCOM is up to 0.05 dB at the lower quantization points up to 0.12 dB at higher points. Compared to Intra H.264, the DIVCOM achieves a higher coding efficiency ranging between 1 kbps and 40 kbps and PSNR gain ranging from 0.06 dB to 0.29 dB for low and intermediate quantization points. However, at high quantization points, the coding efficiency of DIVCOM degrades by 2.8 kbps to 14.4 kbps. The PSNR of DIVCOM is also worse than Intra H.264 by 0.10 dB. The comparison between PDIVCOM and Intra H.264 shows that PDIVCOM achieves a better coding efficiency ranging from 4 kbps to 48 kbps for all quantization points and a PSNR gain ranging from 0.02 dB to 0.32 dB.



**Figure 3.** Rate-distortion (RD) performance graphical evaluation of “Akiyo” test video sequence with frame rate of 15 Hz and GOP 2.

#### 4.2.3. “Foreman” Video Sequence

Table 6 presents the experimental results of the “Foreman” video sequence. It can be seen that DIVCOM can achieve a PSNR improvement of between 0.0043 dB and 0.03 dB over DIS, however, in some cases, DIVCOM gives a worse PSNR than DIS. Additionally, DIVCOM also shows improvement in coding efficiency of between 1.19 kbps to 6.4 kbps over DIS. On the other hand, even though PDIVCOM can achieve a PSNR gain of up to 0.021 dB over DIS, in most cases it performs worse than DIS due to the reconstruction procedure. The coding efficiency of PDIVCOM is better than DIS by between 0.34 kbps to 4.1 kbps but PDIVCOM does not perform as well as DIVCOM.

Table 7 presents the results for the Wyner–Ziv (WZ) coder part for decoding the WZF of the “Foreman” video sequence. For DIVCOM, the average PSNR gain per WZF over DIS ranges from 0.0089 dB to 0.0627 dB while for PDIVCOM, the gain can reach up to 0.0434 dB. The results also indicate that DIVCOM achieves an average channel coding efficiency of between 174 bits to 934 bits, while for PDIVCOM, the coding efficiency ranges from 45 bits to 594 bits. However, the visual quality of both DIVCOM and PDIVCOM is worse than DIS due to the degradation of a few frames during the reconstruction process.

**Table 6.** Visual quality and rate performance evaluation of different codecs for “Foreman” video sequence at frame rate of 15 Hz.

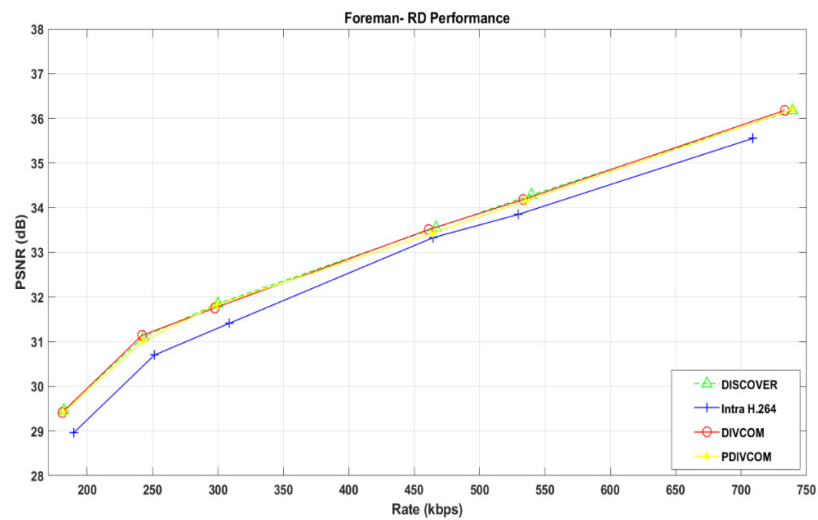
Video Sequence	$Q_m$	Average PSNR (dB)			Coding Rate (kbps)		
		DIS	DIVCOM	PDIVCOM	DIS	DIVCOM	PDIVCOM
Foreman	1	29.467	29.4149	29.4215	182.0371	180.8467	181.7236
	2	31.111	31.1412	31.0124	243.3101	241.8110	242.8442
	3	31.8513	31.7571	31.8224	299.8809	297.5244	298.6641
	4	33.5509	33.5095	33.4341	466.7505	460.9893	464.5913
	5	34.2864	34.1826	34.1587	539.6836	533.3032	535.6260
	6	36.1753	36.1796	36.1962	739.2827	733.3574	739.5210

**Table 7.** Coding efficiency analysis of proposed novel online CNM for “Foreman” video sequence.

Video Sequence	$Q_m$	Average Channel Decoding Rate per WZF (bits)			Average PSNR Gain per WZF Compared to DIS (dB)	
		DIS	DIVCOM	PDIVCOM	DIVCOM	PDIVCOM
Foreman	1	7452	7278	7407	−0.1003	−0.0922
	2	8229	8010	8161	0.0627	−0.2043
	3	12,448	12,103	12,270	−0.1939	−0.06
	4	23,925	23,083	23,610	−0.0857	−0.2419
	5	28,260	27,326	27,666	−0.2149	−0.02645
	6	38,148	37,281	38,183	0.0089	0.0434

Figure 4 shows the RD performance for the “Foreman” test sequence. The graph indicates that both DIVCOM and PDIVCOM achieve better coding efficiency at all quantization points. However, in terms of visual quality, both codecs generally performed worse than DIS, as can be seen from the PSNR values. The PSNR performance degradation of DIVCOM is up to 0.1 dB, while for PDIVCOM, the maximum loss of 0.21 dB. By evaluating the plot for DIVCOM, it is noticed that for lower to middle quantization points, the coding performance of the codec is better than DIS by between 1.19 kbps and 2.36 kbps. This coding efficiency can improve up to 6.38 kbps at higher quantization points. Similarly, PDIVCOM achieves a coding efficiency of between 0.314 kbps and 1.2168 kbps at lower and middle quantization points, and up to 4.0576 kbps at higher quantization points. Therefore, both DIVCOM and PDIVCOM outperformed DIS in terms of coding efficiency but performed worse in terms of visual quality. At low and intermediate quantization points, the bit rate saving achieved by DIVCOM ranges from 3.6 kbps to 8.7 kbps compared to Intra H.264. However, at higher quantization points, DIVCOM shows a higher bit rate ranging from 3.6 kbps to 24.5 kbps. DIVCOM achieves a PSNR of around 0.45 dB to 0.63 dB for all quantization points. The comparative evaluation of PDIVCOM and Intra H.264 shows that the PDIVCOM achieves a bit rate saving ranging from 7.9 kbps to 10.4 kbps for low to intermediate quantization points. However, at high quantization points the coding rate of PDIVCOM is higher than Intra H.264 by up to 30 kbps is noticed. The PSNR gain of PDIVCOM over Intra H.264 ranges from 0.46 dB to 0.69 dB for all quantization points.





**Figure 4.** Rate-distortion (RD) performance graphical evaluation of “Foreman” test video sequence with frame rate of 15 Hz and GOP 2.

#### 4.2.4. “Hall” Video Sequence

Table 8 presents the experimental results of the “Hall” video sequence. It can be seen that both DIVCOM and PDIVCOM achieved a higher average PSNR compared to DIS. When we analyze on a frame-by-frame basis, it is found that for most of the frames a slight PSNR gain is achieved by PDIVCOM, but for some frames the PSNR is reduced. Overall, the average PSNR gain of PDIVCOM over DIS is between 0.005 dB to 0.026 dB. PDIVCOM also achieved an improvement in average coding efficiency by between 0.32 kbps to 4.129 kbps over DIS. Similarly, DIVCOM also shows a variance in the PSNR gain over DIS on a frame-to-frame basis, but the overall average PSNR gain is 0.026 dB. The improvement in coding efficiency of DIVCOM over DIS is in the range of 1.17 kbps to 6.354 kbps. Both DIVCOM and PDIVCOM surpass DIS in terms of coding efficiency, with DIVCOM performing relatively better than PDIVCOM.

**Table 8.** Visual quality and rate performance evaluation of different codecs for “Hall” video sequence for frame rate 15 Hz.

Video Sequence	Q <sub>m</sub>	Average PSNR (dB)			Coding Rate (kbps)		
		DIS	DIVCOM	PDIVCOM	DIS	DIVCOM	PDIVCOM
Hall	1	30.1501	30.1438	30.1489	170.1616	168.9883	169.845
	2	32.87	32.8393	32.8254	243.5151	242.0322	242.8735
	3	34.5575	34.5309	34.5463	320.8701	318.9966	319.6396
	4	35.3398	35.3654	35.3581	389.0669	386.2036	387.0825
	5	36.1409	36.1250	36.1458	445.9565	441.4385	442.5024
	6	36.9788	36.9446	37.0051	533.2949	526.9409	529.1656

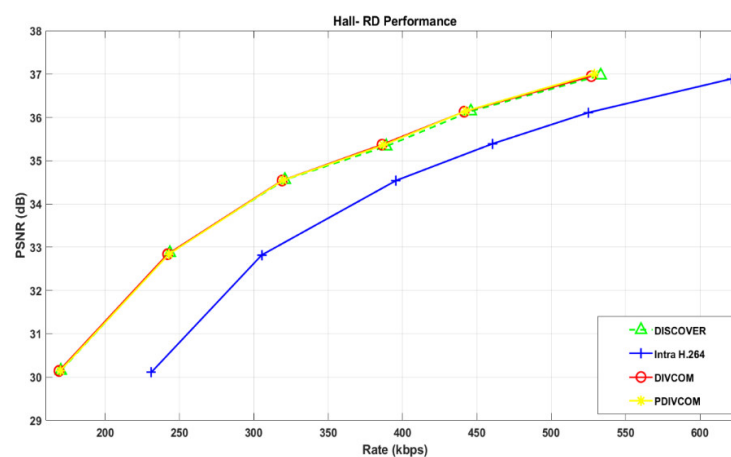
Table 9 presents the results for the Wyner–Ziv (WZ) coder part for decoding WZF. For the “Hall” video sequence, DIVCOM achieved an average PSNR gain of up to 0.053 dB over DIS for the reconstructed WZF, while PDIVCOM achieved an average PSNR gain in the range of 0.0102 dB to 0.0545 dB. The average channel decoding rate results indicate that DIVCOM achieved an average channel coding efficiency of between 46 bits and 929 bits while the coding efficiency of PDIVCOM ranges from 171 bits to 604 bits. Although both DIVCOM and PDIVCOM resulted in poorer visual quality for some of the quantization

points due to degradation in some of the frames during reconstruction, they achieved higher coding efficiency than DIS.

**Table 9.** Coding efficiency evaluation of proposed novel online CNM for “Hall” video sequence.

Video Sequence	$Q_m$	Average Channel Decoding Rate per WZF (bits)			Average PSNR Gain per WZF Compared to DIS (dB)	
		DIS	DIVCOM	PDIVCOM	DIVCOM	PDIVCOM
Hall	1	4467	4421	4296	−0.0131	−0.0023
	2	5123	5029	4906	−0.0109	0.0179
	3	6252	5978	6072	−0.0549	−0.023
	4	11,448	11,029	11,158	0.0531	0.0381
	5	13,081	12,420	12,576	−0.0331	0.0102
	6	17,003	16,074	16,399	−0.0708	0.0545

Figure 4 shows the RD performance for the “Hall” test sequence. This visual representation shows that both DIVCOM and PDIVCOM outperform DIS in coding efficiency. Further analysis shows that DIVCOM outperforms DIS in coding efficiency by a range of 1.17 kbps to 1.87 kbps for lower quantization points and up to 6.354 kbps at higher quantization points. The graph also indicates that PDIVCOM achieves better coding efficiency compared to DIS. For lower and middle quantization points, the coding efficiency of 0.317 kbps to 1.231 kbps is recorded for this test sequence, while for higher quantization points, the coding efficiency increases to 4.13 kbps. In terms of PSNR gain, the performance of DIVCOM and PDIVCOM were mixed. DIVCOM outperforms DIS by up to 0.026 dB but can also suffer a PSNR loss of up to 0.03 dB. Similarly, PDIVCOM outperforms DIS by up to 0.0139 dB but can also suffer a PSNR loss of up to 0.06 dB. The WZF reconstruction process plays a key role in PSNR performance gain. The loss in the performance of the DIVCOM and PDIVCOM for the “Hall” and “Foreman” video sequences is mainly due to the degradation during the reconstruction process. Figure 5 shows that DIVCOM and PDIVCOM outperform the Intra H.264 codec in coding efficiency and PSNR gain for all quantization points. DIVCOM achieves a better coding efficiency ranging from 62 kbps to 94.4 kbps, while its PSNR performance also improves and with a gain of around 0.06 dB. The graph analysis indicates that PDIVCOM achieved a bit rate saving ranging from 61.16 kbps to 91.73 kbps over Intra H.264 for all quantization points. In terms of PSNR gain, PDIVCOM outperforms Intra H.264, by 0.04 dB to 0.12 dB.



**Figure 5.** Rate-distortion (RD) performance of “Hall” video sequence with frame rate of 15 Hz and GOP 2.

#### 4.2.5. Bjøntegaard Delta Performance Evaluation

Table 10 presents the Bjøntegaard metric performance results of DIVCOM, PDIVCOM, and DISCOVER. The codecs' performance results are presented in terms of BD-Rate and BD-PSNR. As reported in Table 10, DIVCOM and PDIVCOM give an improvement in bit rate ranging from 2.46% to 6.13% and 3.47% to 11.37%, respectively, compared to DIS for different video sequences. However, for the Foreman video sequence, the performance of PDIVCOM in terms of bit rate saving is degraded up to 1.33%. The BD-PSNR analysis shows that DIVCOM and PDIVCOM gain 0.05 dB to 0.18 dB and 0.14 to 0.35 dB, respectively, for different video sequences compared to DIS. However, PDIVCOM performance for Foreman is degraded by 0.026 dB compared to DIS.

**Table 10.** Comparison of BD-Rate and BD-PSNR between DIVCOM and PDIVCOM with DIS.

Video	BD-Rate		BD-PSNR	
	DIVCOM vs. DIS	PDIVCOM vs. DIS	DIVCOM vs. DIS	PDIVCOM vs. DIS
Coastguard	−6.13	−6.27	0.14	0.14
Akiyo	−3.46	−11.37	0.113	0.35
Foreman	−1.49	1.33	0.05	−0.026
Hall	−2.46	−3.47	0.18	0.17

## 5. Conclusions and Future Research Work

In DVC, a suitable and accurate correlation noise model plays a crucial role in improving the RD performance and coding efficiency. As in DVC, neither WZF is available at the decoder, nor is the estimated SI of the corresponding WZF available at the encoder. Therefore, online estimation of the CNM and its parameters is quite tricky, especially when the motion vector is not estimated at the decoder such as in [15]. The proposed novel online CNM approach is suitable for such a scenario and accurately calculates the error distribution and makes the coding of the codec highly efficient. Higher coding efficiency and PSNR gain lead to better RD performance. The DIVCOM codec achieves coding efficiency up to 8.05 kbps, and PSNR ranges from 0.0245 dB to 0.18 dB compared to the DIS codec. The PDIVCOM achieves a coding efficiency of up to 10.9 kbps, and PSNR ranges from 0.019 to 0.17 dB compared to DIS.

Even so, there is still room for improvement. During the frame-by-frame analysis, a higher coding rate is observed at some frames of every video sequence. A closer analysis determined that for some frames, the variance was around 0.1, and the channel-coding rate achieved under such conditions was too high. Along with the high channel-coding rate, all errors in bands of such frames were not completely corrected. By further investigating those frames at band level, a higher bit rate is also noticed for most bands of those frames. The CNM parameter ( $\alpha_f$ ) estimation process can further be enhanced by improving the mathematical formulation by considering the variance. This could potentially improve the coding efficiency. Therefore, rectifying the CNM parameter at the band level enhances the coding rate and the overall RD performance. For band-level CNM, we may send more than one sample of coefficient values of a band instead of the mean of coefficients of a band.

**Author Contributions:** Conceptualization, S.K., N.B. and V.J.; methodology, S.K., N.B. and V.J.; validation, N.B., V.J. and M.A.H.; writing—original draft preparation, S.K.; writing—review and editing, N.B., V.J. and M.A.H.; supervision, N.B., V.J. and M.A.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by Institute of Health and Analytics, Universiti Teknologi PETRONAS.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Sehairi, K.; Benbouchama, C.; El Houari, K.; Fatima, C. A Real-Time Implementation of Moving Object Action Recognition System Based on Motion Analysis. *Indones. J. Electr. Eng. Inform.* **2017**, *5*, 44–58.
2. Ukrit, M.F.; Suresh, G. Super-Spatial Structure Prediction Compression of Medical. *Indones. J. Electr. Eng. Inform.* **2016**, *4*, 126–133. [[CrossRef](#)]
3. Deligiannis, N.; Munteanu, A.; Wang, S.; Cheng, S.; Schelkens, P. Maximum likelihood Laplacian correlation channel estimation in layered Wyner-Ziv coding. *IEEE Trans. Signal Process.* **2013**, *62*, 892–904. [[CrossRef](#)]
4. Wang, W.; Zhu, J.; Zhang, S.; Zhou, W. Tradeoff between compression ratio and decoding delay of distributed source coding for uplink transmissions in machine-type communication. *Int. J. Distrib. Sens. Netw.* **2018**, *14*, 1550147718787109. [[CrossRef](#)]
5. Yang, J.; Qing, L.; He, X.; Hua, L.; Rong, S. A Fast DVC to HEVC Transcoding for Mobile Video Communication. In Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 March 2019; pp. 505–509.
6. Dufaux, F.; Gao, W.; Tubaro, S.; Vetro, A. Distributed video coding: Trends and perspectives. *EURASIP J. Image Video Process.* **2010**, *2009*, 508167. [[CrossRef](#)]
7. Guo, M.; Lu, Y.; Wu, F.; Li, S.; Gao, W. Distributed video coding with spatial correlation exploited only at the decoder. In Proceedings of the 2007 IEEE International Symposium on Circuits and Systems, New Orleans, LA, USA, 27–30 May 2007; pp. 41–44.
8. Jun, D. Distributed video coding with adaptive two-step side information generation for smart and interactive media. *Displays* **2019**, *59*, 21–27. [[CrossRef](#)]
9. Van, X.H.; Ascenso, J.; Pereira, F. HEVC backward compatible scalability: A low encoding complexity distributed video coding based approach. *Signal Process. Image Commun.* **2015**, *33*, 51–70.
10. Artigas, X.; Ascenso, J.; Dalai, M.; Klomp, S.; Kubasov, D.; Oualet, M. The DISCOVER codec: Architecture, techniques and evaluation. In Proceedings of the Picture Coding Symposium (PCS'07), Lisboa, Portugal, 7–9 November 2007.
11. Taheri, Y.M.; Ahmad, M.O.; Swamy, M. A joint correlation noise estimation and decoding algorithm for distributed video coding. *Multimed. Tools Appl.* **2018**, *77*, 7327–7355. [[CrossRef](#)]
12. Slepian, D.; Wolf, J. Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory* **1973**, *19*, 471–480. [[CrossRef](#)]
13. Wyner, A.; Ziv, J. The rate-distortion function for source coding with side information at the decoder. *IEEE Trans. Inf. Theory* **1976**, *22*, 1–10. [[CrossRef](#)]
14. Aaron, A.; Zhang, R.; Girod, B. Wyner-Ziv coding of motion video. In Proceedings of the Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2002; pp. 240–244.
15. Khursheed, S.; Jeoti, V.; Badruddin, N.; Hashmani, M.A. Low complexity Phase-based Interpolation for side information generation for Wyner-Ziv coding at DVC decoder. In Proceedings of the 2020 12th International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Porto, Portugal, 20–22 July 2020.
16. Guillemot, C.; Pereira, F.; Torres, L.; Ebrahimi, T.; Leonardi, R.; Ostermann, J. Distributed monoview and multiview video coding. *IEEE Signal Process. Mag.* **2007**, *24*, 67–76. [[CrossRef](#)]
17. Fang, Y. Crossover probability estimation using mean-intrinsic-LLR of LDPC syndrome. *IEEE Commun. Lett.* **2009**, *13*, 679–681. [[CrossRef](#)]
18. Brites, C.; Ascenso, J.; Pereira, F. Studying temporal correlation noise modeling for pixel based Wyner-Ziv video coding. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; pp. 273–276.
19. Brites, C.; Pereira, F. Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding. *IEEE Trans. Circuits Syst. Video Technol.* **2008**, *18*, 1177–1190. [[CrossRef](#)]
20. Huang, X.; Forchhammer, S. Improved virtual channel noise model for transform domain Wyner-Ziv video coding. In Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 921–924.
21. Huu, T.V.; Huong, T.N.T.; Ngoc, M.N.; HoangVan, X. Improving performance of distributed video coding by consecutively refining of side information and correlation noise model. In Proceedings of the 2019 19th International Symposium on Communications and Information Technologies (ISCIT), Ho Chi Minh City, Vietnam, 25–27 September 2019; pp. 502–506.
22. Wu, Y.-Y.; Cai, R.; Zhang, D.-Y. Improved Rate Allocation Algorithm for DVC without Feedback Channel. In Proceedings of the 2015 International Symposium on Computers & Informatics, Beijing, China, 17–18 January 2015; Atlantis Press: Amsterdam, The Netherlands, 2015; pp. 1081–1088.
23. Cui, L.; Wang, S.; Jiang, X.; Cheng, S. Adaptive distributed video coding with correlation estimation using expectation propagation. *Proc. SPIE Int. Soc. Opt. Eng.* **2012**, *8499*, 1380075.
24. Van, X.H.; Ascenso, J.; Pereira, F. Correlation modeling for a distributed scalable video codec based on the HEVC standard. In Proceedings of the 2014 IEEE 16th International Workshop on Multimedia Signal Processing (MMSP), Jakarta, Indonesia, 22–24 September 2014.

25. Deligiannis, N.; Barbarien, J.; Jacobs, M.; Munteanu, A.; Skodras, A.; Schelkens, P. Side-information-dependent correlation channel estimation in hash-based distributed video coding. *IEEE Trans. Image Process.* **2011**, *21*, 1934–1949. [[CrossRef](#)]
26. Cai, R.; Zhang, D.Y. Hybrid Distributed Correlation Noise Model and Parameter Estimation. *Appl. Mech. Mater.* **2015**, *752–753*, 1110–1115. [[CrossRef](#)]
27. Van Luong, H.; Huang, X.; Forchhammer, S. Parallel iterative decoding of transform domain Wyner-Ziv video using cross bitplane correlation. In Proceedings of the 2011 18th IEEE International Conference on Image Processing, Brussels, Belgium, 11–14 September 2011; pp. 2633–2636.
28. Wadhwa, N.; Rubinstein, M.; Durand, F.; Freeman, W.T. Phase-based video motion processing. *ACM Trans. Graph. TOG* **2013**, *32*, 80. [[CrossRef](#)]
29. Meyer, S.; Sorkine-Hornung, A.; Gross, M. Phase-based modification transfer for video. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 633–648.
30. Portilla, J.; Simoncelli, E.P. A parametric texture model based on joint statistics of complex wavelet coefficients. *Int. J. Comput. Vis.* **2000**, *40*, 49–70. [[CrossRef](#)]
31. Simoncelli, E.P.; Freeman, W.T. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In Proceedings of the Proceedings International Conference on Image Processing, Washington, DC, USA, 23–26 October 1995; pp. 444–447.
32. Simoncelli, E.P.; Freeman, W.T.; Adelson, E.H.; Heeger, D.J. Shiftable multiscale transforms. *IEEE Trans. Inf. Theory* **1992**, *38*, 587–607. [[CrossRef](#)]
33. Meyer, S.; Wang, O.; Zimmer, H.; Grosse, M.; Sorkine-Hornung, A. Phase-based frame interpolation for video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1410–1418.
34. Varodayan, D.; Aaron, A.; Girod, B. Rate-adaptive codes for distributed source coding. *Signal Process.* **2006**, *86*, 3123–3130. [[CrossRef](#)]
35. Lam, E.Y.; Goodman, J.W. A mathematical analysis of the DCT coefficient distributions for images. *IEEE Trans. Image Process.* **2000**, *9*, 1661–1666. [[CrossRef](#)] [[PubMed](#)]