Received 16 November 2020; revised 26 February 2021 and 7 April 2021; accepted 8 April 2021. Date of current version 13 May 2021.

Digital Object Identifier 10.1109/OJCAS.2021.3073252

Improved Deep Distributed Light Field Coding

M. UMAIR MUKATI [1], MILAN STEPANOV [2] (Graduate Student Member, IEEE), GIUSEPPE VALENZISE [2] (Senior Member, IEEE), SØREN FORCHHAMMER [1] (Member, IEEE), AND FRÉDÉRIC DUFAUX [2] (Fellow, IEEE)

[1]DTU Fotonik, Technical University of Denmark, 2800 Lyngby, Denmark

[2]Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190 Gif-sur-Yvette, France

This article was recommended by Associate Editor R. Hamzaoui.

CORRESPONDING AUTHOR: M. STEPANOV (e-mail: milan.stepanov@l2s.centralesupelec.fr)

This work was supported by the EU's H2020 ITN Programme, through the MSCA Grant under Agreement 765911 (RealVision).

(M. Umair Mukati and Milan Stepanov contributed equally to this work.)

ABSTRACT Light fields enable increasing the degree of realism and immersion of visual experience by capturing a scene with a higher number of dimensions than conventional 2D imaging. On another side, higher dimensionality entails significant storage and transmission overhead compared to traditional video. Conventional coding schemes achieve high coding gains by employing an asymmetric codec design, where the encoder is significantly more complex than the decoder. However, in the case of light fields, the communication and processing among different cameras could be expensive, and the possibility of trading the complexity between the encoder and the decoder becomes a desirable feature. We leverage the distributed source coding paradigm to effectively reduce the encoder's complexity at the cost of increased computation at the decoder side. Specifically, we train two deep neural networks to improve the two most critical parts of a distributed source coding scheme: the prediction of side information and the estimation of the uncertainty in the prediction. Experiments show considerable BD-rate gains, above 59% over HEVC-Intra and 17.45% over our previous method DLFC-I.

INDEX TERMS Deep learning, distributed source coding, light field, uncertainty estimation, view synthesis.

I. INTRODUCTION

In the pursuit of more immersive visual technologies, Light Field (LF) imaging has risen as an exciting solution to capture rich scene information. LF imaging divides traditional image acquisition by separating the light capturing and image formation. More specifically, in traditional cameras, light rays impinging the sensor are accumulated by a pixel surface resulting in the loss of directional information of the light rays. Conversely, LF imaging allows capturing this additional information and consequently, it offers novel post-capture functionalities such as refocusing and aperture adjustment. However, LF imaging also entails a considerable amount of information that needs to be efficiently compressed. A typical LF image captured by LYTRO Illum camera offers only a 0.25-megapixel resolution albeit occupying about 218 megabytes of hard disk space (i.e., 15 × 15 set of views, 10 bit, three colour channels).

Conventional video coding is designed as a hybrid block-based scheme including prediction, transformation, quantization and entropy coding [1]. The inclusion of the prediction at the encoder side is the primary reason for the superior coding performance compared to transform-based coding. This framework fitted to a broadcast scenario is designed to provide efficient decoding at the cost of heavy computation at the encoder. On the contrary, there are scenarios where it is more desirable to have a power-efficient encoder and transfer most of the computation to the decoder side. These scenarios typically include low-power camera systems, for example, in wireless networks or multi-view video entertainment [2].

Distributed Source Coding (DSC) is an alternative coding paradigm which allows shifting the complexity from the encoder to the decoder. The theoretical foundation of DSC is based on the Slepian-Wolf theorem, which states that, under some conditions, two correlated discrete sources can be encoded independently and decoded jointly, with the same rate as if the two sources were jointly encoded [3]. Later, Wyner-Ziv extended this result to the case of lossy coding of two jointly Gaussian sources, where the coding rate is replaced by a Rate-Distortion (RD) function.

DSC has been explored extensively in the domain of video coding, notably with the development of DISCOVER [4] and VISNET II [5] codecs. In practical Distributed Video Coding (DVC) [6] schemes, video frames are divided into two groups: key frames and Wyner-Ziv (WZ) frames. Key frames are encoded using traditional, hybrid coding schemes. Conversely, WZ frames are initially estimated based on the decoded key frames; this Side Information (SI), available at the decoder, is then corrected through channel codes requested from the encoder. Since generating parity bits (e.g., syndromes [7]) is computationally much lighter than the temporal prediction, the complexity cost at the encoder is reduced by decreasing the number of key frames. This framework has been later extended to Distributed Multiview Video Coding (DMVC) [8]. In the setups with a large number of cameras operating in power-constrained environments, DMVC can effectively reduce the complexity of the encoder (eliminating inter-camera dependency and frame buffering) and shift the prediction between neighboring views at to the decoder side [9]. DSC has been applied to LF as well in the preliminary works [10], [11]. However, distributed coding of LF has remained little explored till now.

In our previous work [12] DLFC-I, we propose replacing a typical optical flow-based prediction scheme with a learning approach to generate high-quality estimates of WZ views while considerably reducing the complexity of the encoder. In this work, we build upon our previous work [12] by further leveraging deep learning approaches for better estimation of SI in the distributed coding scenario. More precisely, we improve the view synthesis performance by considering different arrangements of the reference view and we propose a deep learning-based approach for the estimation of the residual signal. Our contributions are as follows:

- Comparison of four arrangements of reference views, more specifically *Corner*, *Cross*, *Corner-In* and *Cross-In*,
- Comparison of three loss functions for the improvement of view synthesis performance when the reference views are distorted due to HEVC coding,
- A deep learning architecture for the estimation of the residual signal.

Experiments show significant gains of the proposed distributed light field coding scheme compared to the conventional coding tools (operating at similar complexity at the encoder side).

This paper is structured as follow. Section II describes related work, including the coding of different visual modalities using DSC, and deep learning-based view synthesis approaches. In Section III, we explain our proposed variations for the view synthesis network as well as the architecture for uncertainty modelling. Section IV presents the results of the proposed scheme and the comparison with state-of-the-art methods and the conventional coding tools. Finally, Section V concludes the work.

II. RELATED WORK

We divide the related work into three parts: distributed source coding, view synthesis and uncertainty estimation.

A. DISTRIBUTED SOURCE CODING FOR LIGHT FIELDS

DSC was initially used for LF coding by [10], where WZ views are synthesized at the decoder using a geometry-based image rendering from the available key views. To achieve a higher RD performance, the transform domain WZ coding is adopted to exploit better the spatial correlation in [11]. A DMVC approach is proposed in [13]. It generates multiple SIs utilizing temporal and inter-view redundancies. Additionally, a robust fusion method is employed by fusing likelihoods estimated from each SI. The approach can be adapted for light field structures by substituting one angular dimension in place of temporal dimension [12]. PhiCong *et al.* [14] utilize an adaptive strategy to skip WZ decoding process if the synthesized view at the decoder is estimated to have a minimum quality to avoid transmitting bits for that particular view. In order to use existing DVC tools, in [14] the LF views are first downsampled to QCIF resolution and then converted to a pseudo video sequence by utilizing a so-called Hybrid scanning order. Mukati *et al.* [12] propose to use a view synthesis-based approach to synthesize light field views at the decoder utilizing only four key views picked from the four corners of the LF in order to reduce the encoding complexity radically. The results show that leveraging high-quality synthesized views provide competitive RD performance compared to the state-of-the-art DMVC approach [13].

B. VIEW SYNTHESIS

The goal of view synthesis is to generate a novel view from a given set of reference views. Recently, with the wide-spread use of deep learning tools, emerging view synthesis methods allowed the generation of higher-quality views from sparser input sets. Kalantari *et al.* [15] present the first work on view synthesis based on deep learning. The authors follow the traditional scheme for view synthesis, whereas the scheme is factorized into the disparity estimation part, which provides disparity map estimation used to warp reference images, and merging of the warped referenced images. They propose a network which consists of two sequential networks: the disparity network and the colour network. The disparity network takes corner views of a light field image and the novel position of the view to be synthesized. Then, it estimates the disparity of the novel view with respect to the input views. The reference views are then backwarped to obtain the estimates of the novel view and merged by the colour network to obtain the final estimate. Srinivasan *et al.* [16] tackle the problem of estimating the entire light field image from a single image. In particular, the authors estimate the disparity of each pixel in the image and backward warp the input view using the estimated disparity maps to generate a Lambertian light field image. Then, they compensate for the errors due to the occlusions and non-Lambertian effects by estimating these distortions using an additional network. Finally, the proposed framework allows estimating accurate disparity maps in an unsupervised manner by imposing consistency among different maps. Although the work yields interesting results, unsurprisingly, the quality of synthesized views deteriorates considerably when moving away from the centre view. More recently, Navarro and Sabater [17] propose a novel view synthesis approach inspired by these two approaches. The authors estimate a novel view from the corner views as done in Kalantari *et al.* [15], but they also estimate a disparity map of each corner view and merge warped corners using the weights estimated by a selection network. The approach provides superior performance compared to other state-of-the-art approaches and has the potential to operate on wider-baseline light fields.

C. CORRELATION NOISE MODELLING

Accurate SI noise modelling is another important aspect that influences the coding performance as it indicates the reliability of the prediction to an iterative decoder such as LDPCA. In DSC, the correlation noise is generally modelled by a Laplacian distribution. The authors in [18], explore the modelling of the correlation noise at different granularity levels and conclude that a higher granularity level translates to better RD performance, suggesting that the pixel-level and coefficient-level perform best in an offline mode for Pixel-Domain WZ and Transform-Domain WZ, respectively. In online mode, the modelling is done adaptively based on the local intensity variation utilizing motion compensated residuals at different granularity levels, e.g., frame-level, band-level and coefficient-level. In [19], the estimated residual is divided into different classes for each frequency band depending on the estimated residual energy for each block and the Laplacian parameter is found using pre-calculated values in a lookup table. In [20], the Previously Decoded Bands (PDB) are used to improve the noise model by classifying the subsequent residual into two categories. Additionally, a noise residue refinement step updates the noise residual after each band is decoded. In [21], the residual frame is clustered into different classes using Fuzzy C Means based on the residual energy. Contrary to [20], it utilizes all the decoded frequency bands for improved noise modelling.

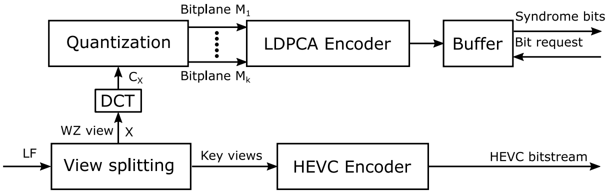
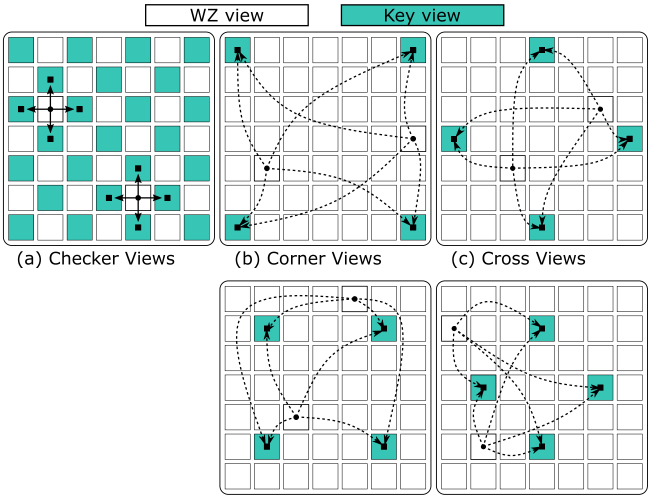


FIGURE 1. Block diagram of transform-domain Wyner-Ziv encoder.



(a) Checker Views (b) Corner Views (c) Cross Views
(d) Corner-In Views (e) Cross-In Views

FIGURE 2. View splitting modes.

III. PROPOSED METHOD

In this section, we describe the proposed Distributed Light Field Coding (DLFC) scheme. In our previous work [12], we have utilized the view synthesis approach, proposed by Navarro and Sabater [17], for the prediction of WZ views. Here, we extend our previous work by considering an improvement of the SI generation whose quality directly correlates with the performance of the coding scheme. To this extent, we explore various modifications in the view synthesis scheme to obtain better prediction across different bitrates and propose a deep learning scheme to estimate the uncertainty of our prediction.

First, we give an overview of the DLFC scheme. Then, we describe a set of enhancements to view synthesis training for improved prediction. Next, we summarize noise modelling in DLFC and propose a learning-based scheme to estimate it. We conclude the section with the description of the training procedure.

A. DISTRIBUTED LIGHT FIELD COMPRESSION

The proposed distributed light field coding scheme is based on transform domain WZ coding with feedback channel [6].

The encoder is presented in Fig. 1. It takes an LF image and extracts and divides views into two sets: key views and WZ views. We select four reference views of an LF image as key views according to one of the four arrangements shown in Figs. 2 (b–e) and process them by a conventional coding

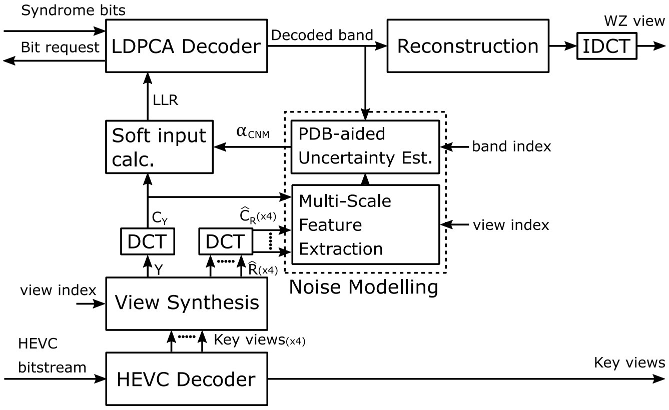


FIGURE 3. Block diagram of transform-domain Wyner-Ziv decoder.

tool, while the rest of the LF are processed using a computationally more efficient WZ encoder. First, each WZ view is transformed block-wise using the 4 × 4 Discrete Cosine Transform (DCT) [22]. Then, the coefficients are quantized using one of eight proposed quantization matrices [4]. In the final step, the quantized coefficients are divided into bitplanes and independently encoded using a Low-Density Parity Check Accumulate (LDPCA) encoder [7]. The computed syndrome bits of each bitplane are stored in the buffer together with 8-bit Cyclic Redundancy Check (CRC).

At the decoder, key views are conventionally decoded and provided to the SI generation block. The role of the SI block is to estimate WZ view, Y, as well as its corresponding residual signal, $\hat{R}$. The SIs are then transformed using the 4 × 4 DCT, resulting in coefficients C_Y and $C_{\hat{R}}$ respectively. The noise modelling block considers Y as a noisy version of the original WZ view and utilizes residual coefficients $C_{\hat{R}}$ for Correlation Noise Modelling (CNM) using the Laplacian distribution. The estimated distribution's parameters α_{CNM} and the prediction coefficients C_Y are provided to the soft input estimation block (and together with the information from the PDBs) used to calculate the bitwise conditional probabilities for each bitplane (soft input). In order to decode bitplanes, the LDPCA decoder needs part of the accumulated syndrome bits from the encoder and the estimated soft input. Using the "message passing algorithm" [23] the decoder iteratively computes the source bits. Upon convergence or pre-defined number of iterations, the procedure stops, and the decoder computes the syndrome bits from the estimated source bits. If the computed syndrome bits matches the received syndrome stream and passes CRC checksum test, then the decoding is considered as successful. Otherwise, the decoder requests more bits from the encoder. After successful decoding of all bitplanes, the quantization intervals of a WZ view are obtained. In the final step, the WZ view is reconstructed using the maximum likelihood approach utilizing estimated Laplacian distribution and decoded quantization intervals [24]. The reconstructed view is transformed back to the pixel domain using the inverse DCT.

B. VIEW SYNTHESIS

1) BASELINE SYNTHESIS APPROACH

For the sake of completeness, we briefly describe the view synthesis approach used in our previous work [12]. For a more detailed description, the reader may also consult [17].

The view synthesis approach consists of three sequential networks: the feature extraction network, the disparity estimation network and the selection network. The feature extraction network takes corner views (of an LF image) and the angular position of a novel view and extracts relevant information for the following stage. The disparity estimation network takes extracted features and the position of the novel view and estimates the disparity map of the novel view with respect to each corner view. Then, the corner views are warped following the estimated disparity maps and finally merged in the final estimation as a weighted sum with weights obtained by the selection network. The network is optimized using a two-parts loss $\mathcal{L}_{l1-grad}$ which includes the L1 loss between the original image texture I and the synthesized image texture Y and the L1 loss of the gradients of the two textures:

$$\mathcal{L}_{l1-grad} = \|I - Y\|_1 + \frac{1}{2}\|\nabla I - \nabla Y\|_1. \tag{1}$$

2) CHOICE OF REFERENCE VIEWS

In the coding of LF images using traditional coding tools, such as HEVC, much effort has been put into finding an optimal coding order, and it has been shown that the prediction from closer views provides better performance [25]. A typical configuration for view synthesis tasks includes a set of *Corner* views in an LF image as they capture the widest field of view. In this paper, we consider three more arrangements of the four reference views as shown in Figs. 2 (c–e) to utilize the one for SI generation, which provides the best prediction quality.

3) LOSS FUNCTION

Furthermore, we evaluate two loss functions which could increase the performance of view synthesis, especially with the decrease in the quality of reference views. More precisely, we consider a perceptual loss based on high-level feature maps of a deep neural network VGG utilized for the image classification task [26] and a loss which includes uncertainty modelling of the prediction [27].

The early layers of the VGG network give a response highlighting low-level features of the input, while the deeper layers capture higher semantic information [28]. We assume that the inclusion of semantic reasoning will aid the view synthesis network to generalize better in the case of distorted input. We use pre-trained VGG-19, which is available in the Pytorch framework and extract the activations from five layers as it is typically done in the literature [29], [30] to compute the loss:

$$\mathcal{L}_{vgg} = \sum_{l}^{L} \lambda_l \|\Phi_l(I) - \Phi_l(Y)\|_1, \tag{2}$$

where Φ_l denotes the activations inferred from the layer l.

Kendall and Gal [27] propose a loss function that considers the uncertainty in the prediction for the depth regression and semantic segmentation tasks. The loss function can be considered as learned attenuation as it penalizes the samples based on their prediction fidelity and provides a more robust estimation. Although our task does not explicitly regress depth, it highly depends on the estimated disparity maps at the intermediate levels. Moreover, our view synthesis task relies on the selection network to provide (soft) recommendations of the final prediction at the pixel level. Therefore, the robust estimation of the disparities should be beneficial to the final prediction. We add a branch, which estimates uncertainty on a pixel level, to the original network, and feed both estimates, the prediction and the uncertainty, to a loss function defined as a negative logarithm of the likelihood of the Laplacian distribution. Note that it is also possible to select a Gaussian distribution. However, we choose the Laplacian as it is typically used to model the distribution of a residual signal. The Laplacian loss-based version of view synthesis approach is defined as follows:

$$\mathcal{L}_{laplacian} = -\frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{\alpha(n)}{2}\exp^{-\alpha(n)|I(n)-Y(n)|}\right), \quad (3)$$

where N is the total number of pixels in a batch, $\alpha(n)$ is the predicted Laplacian distribution parameter, and $I(n)$ and $Y(n)$ are ground truth and predicted pixel values, respectively.

C. CORRELATION NOISE MODELLING

In an offline design process, the residual signal is used to model the correlation noise in the prediction of the WZ view. Typically, the Laplacian distribution offers a reasonable fitting to the distribution of the correlation noise, where the distribution's parameter α_{CNM} should describe the reliability of the prediction. It has been observed that the statistics of the correlation noise vary locally [18]. Therefore, estimating the distribution at the finer level is desirable. As reported in [18], the noise modelling at the finest level, i.e., pixel-level in the pixel-domain WZ or coefficient-level in the transform-domain WZ, offer optimal RD performance.

For example, the model parameter α_{CNM} of each coefficient (u, v) is defined inversely proportional to the absolute coefficients of residual signal $C_R(u, v)$ [18]:

$$\alpha_{CNM}(u, v) = \frac{\sqrt{2}}{|C_R(u, v)|}. \quad (4)$$

Due to the unavailability of the original WZ view at the decoder, the actual correlation noise cannot be used to model the distribution. Instead, the modelling is usually done by substituting the actual residual signal with the difference in the two predictions of the WZ view, as the agreement in the two predictions represents the likelihood of the accuracy in the prediction. This approach can model well the correlation noise in prediction at the coarsest level. As we move towards the finer level, the noise modelling becomes unreliable due to an insufficient number of samples required for accurate modelling as well as the uncertainty in the residual estimation itself. Therefore, several methods have been proposed in the literature for robust correlation noise modelling, e.g., [18], [20].

In our prior work [12], we have used the approach described in [20] for the noise modelling using the estimated residual signal. As for estimating the residual signal, we have used a weighted average of the estimated intermediate residuals corresponding to the four corner views used at the input of the view synthesis method. The intermediate residuals $\hat{R}_i$ are calculated as follows:

$$\hat{R}_i(x, y) = Y(x, y) - W_i(x, y), \quad (5)$$

where Y is the predicted view, and W_i is the warped view corresponding to the reference corner view i. The weight of each intermediate residual signal at the pixel level is assigned with a higher value when its corresponding residual is lower.

We have noted that the RD performance still lacks in performance compared to the case when the original residual is used for noise modelling in the offline process. We propose to leverage a learning-based approach to optimally estimate the residual signal using the predicted WZ view and the warped residuals. In [18], for the robust noise modelling, based on the local variation in the neighbourhood, the variances estimated from coarse-to-fine levels are assigned at the pixel-level. The correlation between models across different bands is also exploited for improved modelling in [31]. We consider both these approaches to design a network that can robustly estimate the residual signal.

1) PROPOSED NETWORK TO MODEL THE RESIDUAL SIGNAL

As our scheme is based on transform domain WZ, the residual is initially transformed to calculate α_{CNM}. The DCT transformation requires a signed residual as an input. As the absolute value of the transformed residual $|C_R|$ is utilized in (4), we directly estimate $|C_R|$ using the network. In this way, we can calculate the absolute value of the transformed residual signal directly and simplify our prediction.

The proposed network consists of two parts that estimate the absolute coefficients of the residual signal in two steps. These two parts are detailed in Tables 1 and 2, respectively. The first network extracts multi-scale spatial features from the synthesized view and the estimated residual signals. The statistics of the residual signal remains mostly constant across all the frequency bands. Utilizing them will help the network to generalize well across different datasets and frequency bands. Therefore, the first set of blocks of the network $\mathcal{F}_{INT}$, $\mathcal{F}_{MS}$, $\mathcal{F}_G$ are trained to learn common features across all the bands through weight sharing by utilizing 3D kernels with depth size of 1. It is also important to consider the difference in the properties of the residual signals of different frequency bands. Therefore, we utilize another set of layers in the block $\mathcal{F}_S^b$ that is uniquely trained to process each frequency band b.

TABLE 1. The network architecture of initial residual estimation. k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation.

	Name	k	In	Out	Depth	Act. f.
$\mathcal{F}_{INT}$	Input		7			
	conv0	$3\times3\times1$	7	16	16	ELU
	conv1	$3\times3\times1$	16	32	16	ELU
	conv2	$3\times3\times1$	32	32	16	ELU
	conv3	$3\times3\times1$	32	32	16	ELU
	conv4	$3\times3\times1$	32	32	16	ELU
	Output: F_{int}			32	16	
$\mathcal{F}_{MS}$	Input: F_{int}		32		16	
	conv0	$3\times3\times1$	32	32	16	ELU
	conv1	$3\times3\times1$	32	32	16	ELU
	conv2	$3\times3\times1$	32	16	16	ELU
	conv3	$3\times3\times1$	16	4	16	ELU
	Output: F_3			4	16	
	Input: F_{int}		32		16	
	conv0	$5\times5\times1$	32	32	16	ELU
	conv1	$5\times5\times1$	32	32	16	ELU
	conv2	$5\times5\times1$	32	16	16	ELU
	conv3	$5\times5\times1$	16	4	16	ELU
	Output: F_5			4	16	
	Input: F_{int}		32		16	
	conv0	$7\times7\times1$	32	32	16	ELU
	conv1	$7\times7\times1$	32	32	16	ELU
	conv2	$7\times7\times1$	32	16	16	ELU
	conv3	$7\times7\times1$	16	4	16	ELU
	Output: F_7			4	16	
$\mathcal{F}_{G}$	Input: Concatenate $[\mathcal{F}_3, \mathcal{F}_5, \mathcal{F}_7]$					
	conv0	$3\times3\times1$	12	32	16	ELU
	conv1	$3\times3\times1$	32	32	16	ELU
	conv2	$3\times3\times1$	32	32	16	ELU
	Output: F_g			32	16	
$\mathcal{F}_{S}^{b}$	Input: $F_g\left(b\right)$		32		-	
	conv0	3×3	32	32	-	ELU
	conv1	3×3	32	16	-	ELU
	conv2	3×3	16	4	-	ELU
	conv3	3×3	4	1	-	
	Output: $F_s\left(b\right)$			1	-	

The block $\mathcal{F}_{INT}$ extracts some intermediate features F_{int} in the following way:

$$F_{int} = \mathcal{F}_{INT}\Big(C_Y, C_{\hat{R}_1}, C_{\hat{R}_2}, C_{\hat{R}_3}, C_{\hat{R}_4}, P, Q\Big), \tag{6}$$

where C_Y is the transformed coefficients of the predicted WZ view and $C_{\hat{R}_i}$ is the transform of the estimated residual corresponding to the cross-view i calculated using (5). Additionally, the tensors P and Q consisting of the current view index p and q, respectively, are passed to this layer for the network to learn the view-position dependent features. This results in a 3D input volume with 7 channels. The output F_{int} is then passed to three parallel sets of convolutional layers, $\mathcal{F}_{MS}$, that learn to filter the intermediate features at multiple levels, i.e., with kernels of different receptive fields. These outputs are then concatenated and processed by $\mathcal{F}_G$. Finally, the features specific to each frequency band b are learned by $\mathcal{F}_S^b$:

TABLE 2. The network architecture of refined residual estimation (aided by decoded bands). k denotes the size of convolution kernel, In and Out denote the number of input and output channels and Act. f. denotes the name of activation. In this network each layer is followed by batch normalization.

	Name	k	In	Out	Depth	Act. f.
$\mathcal{D}^b$	Input		17			
	conv0	3×3	17	32	-	ELU
	conv1	3×3	32	64	-	ELU
	conv2	3×3	64	64	-	ELU
	conv3	3×3	64	32	-	ELU
	conv4	3×3	32	32	-	ELU
	conv5	3×3	32	1	-	ELU
	Output: $F_d\left(b\right)$			1	-	
$\mathcal{R}^b$	Input: Concatenate $\left[F_s\left(b\right), F_d\left(b\right)\right]$					
	conv0	3×3	2	32	-	ELU
	conv1	3×3	32	32	-	ELU
	conv2	3×3	32	16	-	ELU
	conv3	3×3	16	4	-	ELU
	conv4	3×3	4	1	-	
	Output: $\beta\left(b\right)$			1	-	

$$F_s(b) = \mathcal{F}_S^b(\mathcal{F}_G(F_3, F_5, F_7)). \tag{7}$$

It should be noted that this network tries to learn the features without exploiting inter-band correlation. It is shown in [20], [21] that there exists some correlation in the residual signals for different frequency bands. Hence, exploiting the correlation utilizing PDBs will improve the residual estimation process.

The second network is composed of two parts. The first part $\mathcal{D}$ processes the PDBs to exploit inter-band correlation. Instead of passing decoded bands to the network, the target residual C_R^q (the difference between the quantized coefficients of the WZ view and the coefficients of the prediction C_Y) of these bands are computed and then provided to the block $\mathcal{D}$:

$$F_d(b) = \mathcal{D}^b\big(C_R^q \cdot M(b), b\big), \tag{8}$$

where $M(b)$ masks out the non-decoded bands in C_R^q. The features $F_d(b)$ and $F_s(b)$ are passed to the second part of this network $\mathcal{R}$ which makes the final prediction $\beta(b)$. The network is trained such that $\beta(b)$ represents the absolute coefficients of the residual which can be used to calculate $\alpha_{CNM}(b)$ for each band b in the following way:

$$\alpha_{CNM}(b) = \frac{\sqrt{2}}{\beta(b)} \tag{9}$$

The LDPCA decoder can only decode the coefficient of a WZ view up to some quantization level; therefore it is intuitive to train a network for the quantized target residual C_R^q. In addition, the estimated residual plays a vital role in the reconstruction part as it is used along with the synthesized view and the decoded bands to find a maximum likelihood solution. We have observed that in this case, the true residual signal C_R, i.e., the difference between unquantized coefficients of the WZ view and the coefficient of the prediction C_Y, results in the optimal reconstruction performance. Hence, two networks are trained for each residual signal.

The second network in the proposed scheme utilizes the quantized decoded bands. The statistics of decoded bands vary from one quantization index to another. To achieve the best performance, the networks are trained for each quantization index M independently. Each layer in the residual estimation network is followed by batch normalization.

D. TRAINING DETAILS

For training, we use the *Flowers* dataset [16] which consists of 3343 images of plants. We select one hundred images for validation and the rest of the dataset for training. At each training iteration, we randomly crop training samples to the spatial size 192 × 192, randomly select the position of the novel view, excluding the positions of the corner views, and augment processed samples by applying gamma correction with the gamma value randomly selected from the range [0.4, 1.0]. We observe the convergence of the model on the validation set wherein we use the full spatial size, select centre views only, and randomly select the gamma value from the range [0.4, 0.5]. We use ADAM optimizer with default parameters and set the batch size to 10.

For training the network for residual estimation, we need to provide the data in the transformed domain. A trained model for the view synthesis network is used, which provides the prediction of the WZ view of spatial size 192 × 192, which, after transformation, results in 48 × 48 spatial resolution. Therefore, the network is trained with batches having 48 × 48 block size for all the inputs. Considering the nature of the residual signal, we have used the Laplacian distribution as the loss function to train the residual estimation networks for coding and reconstruction using $\mathcal{L}_C$ and $\mathcal{L}_R$, respectively.

$$\mathcal{L}_C = \sum_b \log \beta_C(b) + \frac{|C_R(b)|}{\beta_C(b)}, \tag{10}$$

$$\mathcal{L}_R = \sum_b \log \beta_R(b) + \frac{|C_R^q(b)|}{\beta_R(b)}, \tag{11}$$

where β is the variance of the Laplacian distribution estimated at the coefficient-level. The loss functions $\mathcal{L}_C$ and $\mathcal{L}_C$ reach their optimal minima when $\beta_C = |C_R^q|$ and $\beta_R = |C_R|$, respectively.

The networks are trained in Python using PyTorch framework. Each view synthesis network is trained for 300 epochs which takes around 15 hours on GeForce RTX 2080 Ti GPU. Whereas, each residual estimation network is trained for 750 epochs which takes around 37 hours on Tesla V100 GPU.

IV. RESULTS

In this section, we describe the testing conditions and report the performance of the proposed scheme in comparison to relevant state-of-the-art schemes.

A. TEST CONDITIONS

In our previous work [12], we prepared the test set *EPFL-DAN* following the recommendations given by JPEG Pleno [32]. We note that the light fields in the *Flowers* dataset used for training the networks are decoded using the Lytro Power Tool (LPT) [33] and have different characteristics than the light fields in *EPFL-DAN* decoded using the Dansereau's Toolbox [34]. Therefore, our model is likely to perform better on LPT decoded datasets. Thus, we create a new test set *EPFL-LPT* by decoding the lenslets in the *EPFL* dataset [35] using LPT.

For experiments, we use three different datasets, out of which two are decoded using LPT [33] (*California* and *EPFL-LPT* datasets) and one decoded using Dansereau's toolbox [34] (*EPFL-DAN* dataset). The *EPFL-LPT* dataset is composed of 8 LF images, as shown in Fig. 4(i), while the *California* dataset is composed of 30 test LF images used in [15]. The decoded LF images have 14 × 14 set of views of 376 × 541 pixels. The dataset *EPFL-DAN* utilizes the same set of 8 LF images as in *EPFL-LPT* but is decoded using [34]. The resulting LF image provides a 15 × 15 set of views of 434 × 625 pixels. In our experiments, we crop each LF to central 7 × 7 set of views due to noticeable artefacts at peripheral views, which would degrade the view synthesis performance.

B. VIEW SYNTHESIS

In this section, we sequentially analyze the performance of the view synthesis approach based on the variations proposed in Section III-B and utilize the approach that generally performs best in terms of objective quality for the SI generation in the proposed DLFC scheme.

In the first experiment, we compare the performance concerning the arrangements of the four reference views. For each of the four arrangements shown in Figs. 2 (b–e), the view synthesis network is independently trained. Table 3 provides the quantitative analysis for the performance of the view synthesis network for each of the reference views arrangement utilizing the three datasets described earlier. Overall, it can be observed that the cross arrangements performed better across all the datasets. Moreover, since the view synthesis network is trained on LPT datasets, the *Cross* arrangement performs better on the *EPFL-LPT* and *California* datasets. Based on the superiority of *Cross-In* arrangement on the *EPFL-DAN* dataset, it can be deduced that this arrangement generalizes the light field structure better. Generally, it can be observed for the datasets decoded using LPT that significantly higher quality is achieved across different reference view arrangements than the dataset decoded using Dansereau's toolbox, i.e., *EPFL-DAN*. This comparison suggests that the trained models generalize well

TABLE 3. Performance evaluation of four arrangements for view synthesis task across three datasets in terms of PSNR (dB).

Dataset	*Corner*	*Corner-In*	*Cross*	*Cross-In*
California	38.20	38.64	39.07	38.90
EPFL-LPT	39.50	40.62	40.98	40.77
EPFL-DAN	30.65	32.48	32.17	32.65

TABLE 4. Performance evaluation of three loss functions for view synthesis task on Cross arrangement across three datasets in terms of PSNR (dB).

Dataset	$\mathcal{L}_{l1-grad}$	$\mathcal{L}_{vgg}$	$\mathcal{L}_{laplacian}$
California	39.07	38.46	38.12
EPFL-LPT	39.98	39.44	38.96
EPFL-DAN	32.17	32.43	32.49

across different datasets but not across different LF decoding schemes. For the rest of the evaluation, we consider the *Cross* arrangement as the default arrangement for the proposed approach due to its superiority on the LPT decoded datasets. From Table 3, it can also be observed that even though the inward variant of *Corner* arrangement improves performance compared to the original variant, this trend does not repeat in the case of *Cross* view arrangements. We explain this behavior by considering the similarity between reference views and the rest of the light field. Namely, by reducing the distance between the reference views, the prediction quality of the in-between views should increase as they become more correlated. Conversely, the quality of the extrapolated views degrade with increase in their distances from the reference views. Therefore, it would be beneficial to find an optimal set of reference views for which the quality of synthesized in-between views increases while the quality of extrapolated views does not degrade considerably. Based on the results presented in Table 3 it can be noted that a "sweet spot" lies around *Cross* reference arrangement for datasets decoded using LPT and *Cross-In* reference arrangement for *EPFL-DAN* dataset.

Next, we explore two loss functions as proposed in Section III-B. From Table 4, it can be observed that $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ versions underperform compare to the original version $\mathcal{L}_{l1-grad}$ on LPT decoded datasets. On the other hand, the evaluation of the *EPFL-DAN* dataset suggests that some loss functions generalize better than others across different decoding schemes, e.g., $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$. This result motivates to further explore these variants for the distorted inputs, which will be provided to the view synthesis network at the decoder of the proposed DLFC scheme.

Table 5 provides a quantitative evaluation in the case of distorted input views. We also compare three loss functions in the *Cross* arrangement. Comparing Tables 4 and 5, we observe in the case of undistorted inputs that the original loss function $\mathcal{L}_{l1-grad}$ performs better compared to both $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ losses. In the case of distorted input views, we note the same behavior with a small exception in the case of the loss $\mathcal{L}_{laplacian}$ which seems to degrade relative quality between different quality levels less, compared to the two other loss functions.

TABLE 5. Quantitative evaluation of view synthesis approach given distorted *Cross* arrangement reference views from the *EPFL-LPT* dataset in terms of PSNR (dB).

QP	$\mathcal{L}_{l1-grad}$	$\mathcal{L}_{vgg}$	$\mathcal{L}_{laplacian}$
27	37.96	37.71	36.95
32	35.78	35.56	35.12
38	32.61	32.41	32.30
45	28.78	28.63	28.74

TABLE 6. Quantization parameters of the key views corresponding to four quantization indices M = [1, 4, 7, 8] from the set in [4] to have consistent quality of reconstructed views.

Sequence	Q_1	Q_4	Q_7	Q_8
Bikes	41	29	25	22
Danger	41	30	25	22
Desktop	42	29	25	22
Flowers	40	30	25	22
Fountain	42	32	27	23
Friends	40	27	23	20
Stone	38	28	23	21
Vespa	41	28	24	21

Although we can observe better generalization of $\mathcal{L}_{vgg}$ and $\mathcal{L}_{laplacian}$ version across different datasets, these trends do not repeat on the distorted datasets. Therefore, we adopt the version of the network trained using the original loss function $\mathcal{L}_{l1-grad}$ in subsequent experiments.

C. DISTRIBUTED LIGHT FIELD CODING

To analyze the RD performance, we utilize the *EPFL-LPT* dataset. Firstly, the effective resolution of each view is set to 376×544 by zero-padding (governed by 4×4 DCT operation in the transform-domain WZ codec, which demands that the resolution of a view be a multiple of four) as the original resolution is 376×541 pixels. After transformation, each frequency band has an effective resolution of 94×136 pixels. Since the bitplanes for each frequency band are encoded one at a time by the LDPCA encoder, this results in a source code of length 12784 bits. We design LDPCA codes for this length following the procedure described in [7]. Only the luminance channel is used to report the performance.

The four key views are decoded using HEVC Intra decoder (HM reference software, v.16.22, with Range Extension (RExt) mode and Main profile). The RD performance of distributed coding schemes is evaluated at four different RD profiles by selecting quantization matrices from [4] at quantization indices $M = [1, 4, 7, 8]$. To have the same quality key views and WZ views after the reconstruction, the QP parameter in HEVC is selected to match the quality of the reconstructed WZ view for each LF and quantization index, as specified in Table 6.

In this section, we will utilize a naming convention for clarity and designate our proposed approach as Cross-Net in addition to DLFC since *Cross* arrangement of views is utilized and the residual signal is estimated using the network-based approach. To assess the proposed method's RD performance, we conduct ablation studies on variations of the distributed coding scheme. These variations are

Bikes Danger Desktop Flowers

Fountain Friends Stone Vespa

FIGURE 4. Thumbnails of light fields from the EPFL dataset [35].

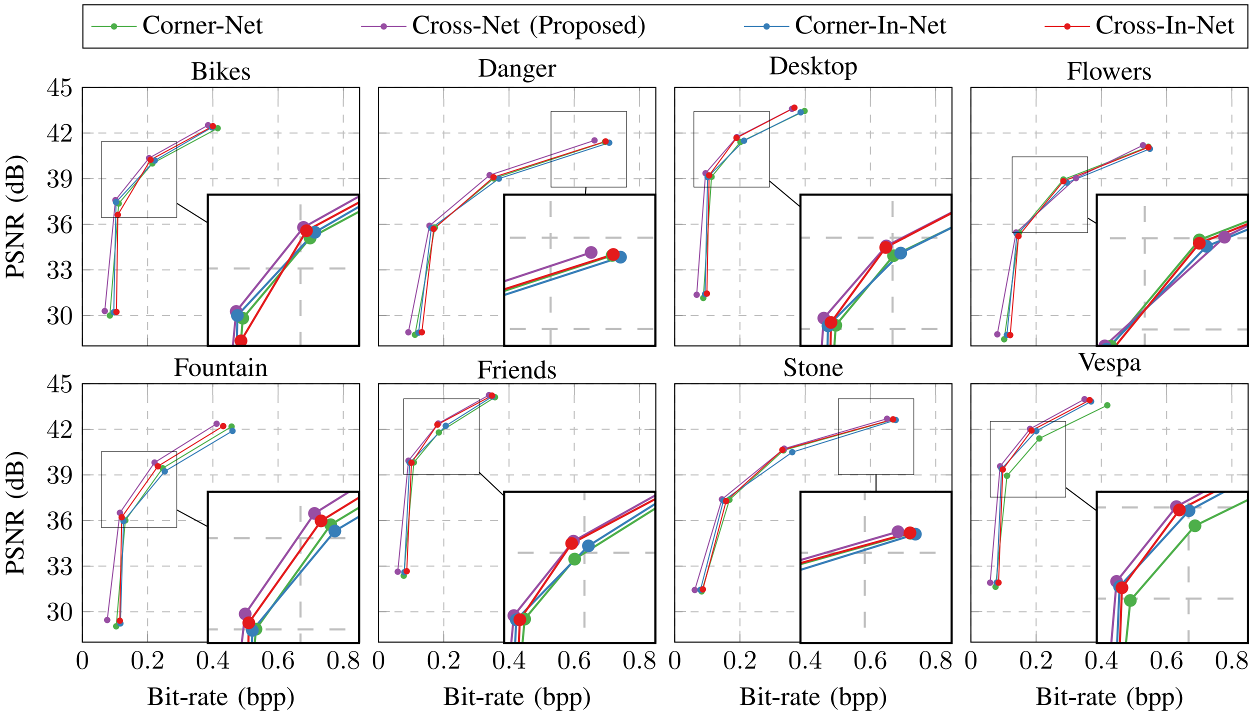


FIGURE 5. RD performance comparison between four different variations of the proposed DLFC scheme based on the arrangements of the reference views shown in Figs. 2 (b–e), at quantization indices $M = [1, 4, 7, 8]$, using PSNR as distortion metric.

obtained using different arrangements of the reference key views and different methods to estimate residual signals.

First, we consider different arrangements of the reference key views as shown in Figs. 2 (b–e). Although the superiority of the *Cross* arrangement of reference views is already proven in the previous section, an RD performance comparison can further establish its supremacy when used alongside the proposed residual estimation network. Fig. 5 plots the RD performance utilizing different reference views arrangements. With the exception of the *Flowers* light field, it can be observed that Cross-Net generally outperforms methods with different reference views arrangement and achieves higher performance at all quantization index values.

Next, we study the effect of utilizing different methods for estimating residual signals in the overall RD performance. The first variation, in this case, can be adopted from our previous work [12], which utilizes weights, calculated based on the four independent residual estimates obtained from each reference view, to estimate the final residual signal. We denote this approach as Cross-Weighted when used alongside the *Cross* arrangement of the reference views. As another variation, we introduce Cross-Ideal, which utilizes

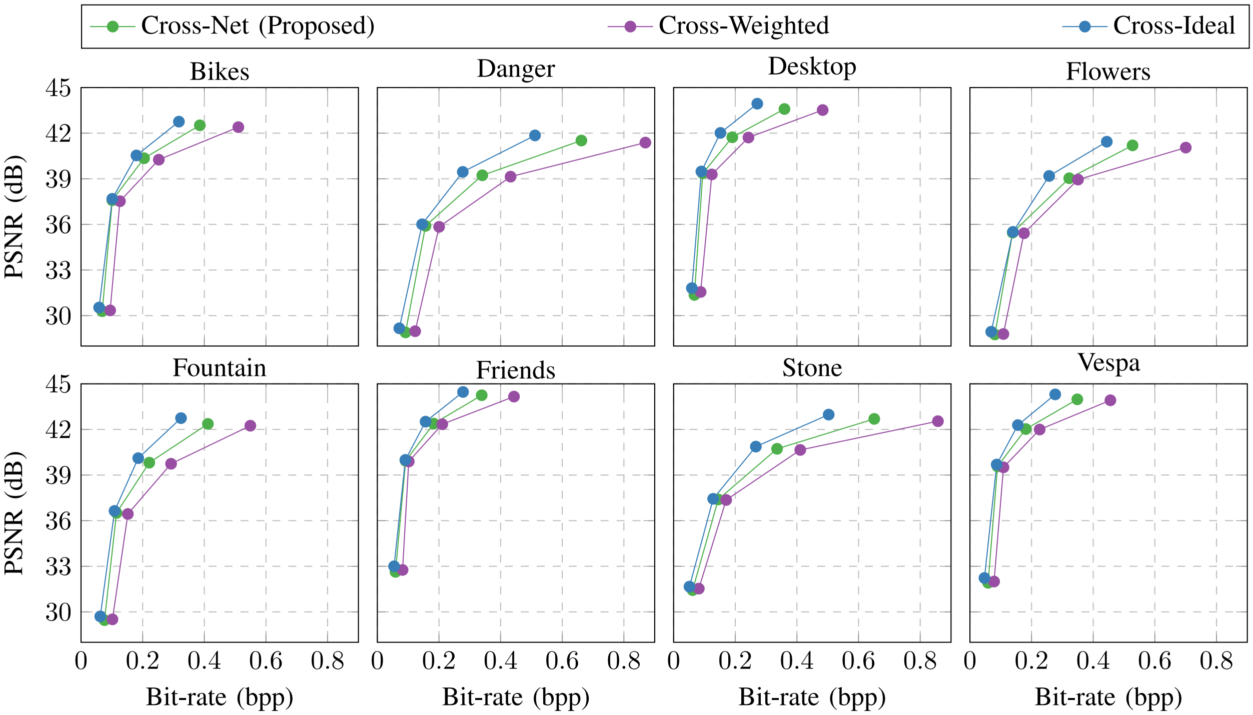


FIGURE 6. RD performance comparison between different variations of the proposed DLFC scheme utilizing three different residual estimation methods, at quantization indices $M = [1, 4, 7, 8]$, using PSNR as distortion metric.

the ideal residual signal to set the upper bound of the achievable performance. Fig. 6 acknowledges the improvement achieved using the network-based approach Cross-Net to estimate the residual signal over Cross-Weighted. However, it can be inferred by looking at Cross-Ideal curves that even with accurate residual estimation, the performance can not surpass the upper bound set by it. Given this situation, we can say that considerable improvement is achieved over Cross-Weighted using Cross-Net.

1) COMPARISON WITH ANCHORS

To evaluate the performance of the proposed DLFC scheme, we compare our performance with two distributed light field coding schemes. Additionally, we provide comparisons with conventional (non-distributed) coding schemes for light field coding.

First, we compare the proposed scheme with our previous work [12], referred to as DLFC-I. This scheme is different from the proposed approach in multiple ways. Primarily, the *Corner* arrangement of the reference views is used. Additionally, the residual signal is estimated through mathematical manipulation of the individual residual signals obtained by subtracting warped corner views from the synthesized view. It further incorporates a strategy to classify the residual signal based on the previously decoded bands to model the Laplacian distribution adaptively. As in [12], we also compare to the DMVC method [13] (referred here as Checker-MultiSI), which presents the state-of-the-art approach in this domain adapted for light field scenario. Here, the views are split in a checkerboard pattern, as shown in Fig. 2, to utilize horizontal and vertical adjacent neighbors of a WZ view for its prediction. Contrary to DMVC, an additional angular dimension is substituted in place of the temporal dimension. Other than the higher encoding complexity, the approach is expected to have a competitive performance as it generates high-quality prediction due to the narrower baseline among the available neighboring views than the *Cross* arrangement of reference views.

For comparison with conventional coding schemes, we select HEVC-Intra as the first anchor to compress all the views independently. The same HEVC configuration is utilized for the key-views coding. Inspired by the comparison provided in [37], we compare our approach with HEVC-NoMotion, which is superior to the former approach because it exploits temporal redundancy like HEVC-Inter, but the motion search range is set to zero. The configuration provided in [37] has been used to configure the HEVC encoder for HEVC-NoMotion. The encoder is provided with the 1-D sequence of LF views as a pseudo video sequence, generated by following a serpentine scanning order. A relevant anchor to compare is the standard light field coding scheme provided by JPEG-Pleno [36]. We compare only to MuLE, i.e., transform-based mode of the reference software, as it has been shown that it is superior compared to the prediction-based mode, i.e., WaSP, on lenslet data [38]. MuLE utilizes a 4D-DCT transform to concentrate the energy of the light field image to a smaller region. This study provides a discussion and comparison of the two coding paradigms, i.e., distributed coding and conventional coding.

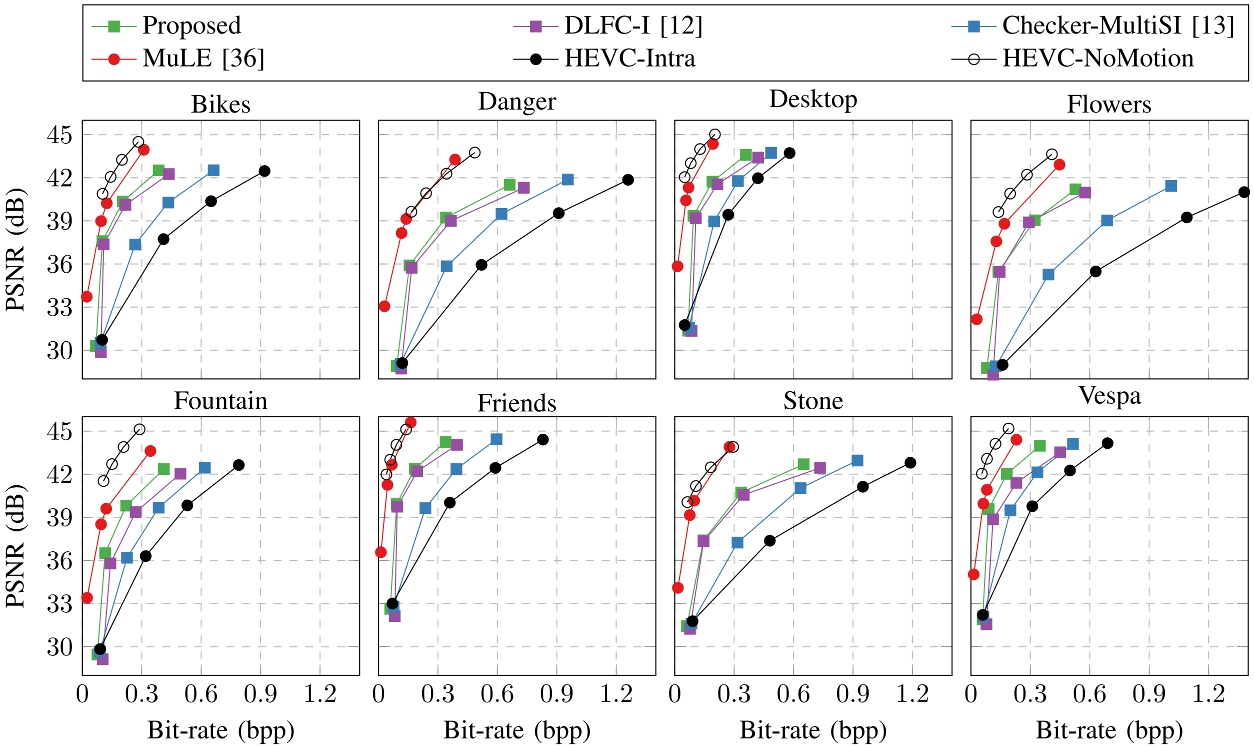


FIGURE 7. RD performance comparison of distributed source coding and conventional coding schemes using PSNR as distortion metric at quantization indices $M = [1, 4, 7, 8]$, whereas, the quantization parameters specified in Table 7 are used for both HEVC plots.

Fig. 7 plots the RD performance of the above-described schemes and the proposed method, utilizing PSNR as a distortion metric. Table 7 quantifies the performance of the distributed coding schemes in comparison to HEVC-Intra using Bjøntegaard measure [39]. It can be observed that the proposed scheme outperforms both distributed coding architectures. The higher RD performance of the proposed approach compared to the DLFC-I can be attributed to the quality gains in the view synthesis approach and the improvement in residual signal estimation using the network-based scheme. In the case of Checker-MultiSI, we would expect higher performance due to the availability of closer reference views for view synthesis. However, it requires half of the views to be encoded using HEVC-Intra, thus reducing the overall RD performance. Quantitatively, our approach achieves 0.96 dB and 4.02 dB gains in BD-PSNR, and 17.45% and 46.66% reduction in BD-Rate, in comparison to DLFC-I and Checker-MultiSI on average, respectively.

From Fig. 7, it can be observed that all the variations of distributed coding significantly outperforms HEVC-Intra due to the high quality of the synthesized views. This is evident by observing that the difference in performance reduces as the distortion increases. With a higher distortion, the compression artefacts become significant in the key views, due to which view synthesis can no longer exploit the common feature points in all the key views. Overall, it can be observed from Table 7 that the distributed coding schemes achieve roughly $50\% - 65\%$ improvement in BD-Rate and 4.5 dB - 6.2 dB gains in BD-PSNR. It may be noticed in our previous work [12] that HEVC-Intra performed comparably to the distributed coding schemes. After emphasizing that the inter-view correlation is exploited in the distributed coding scheme at the decoder, we highlight that the dataset *EPFL-DAN*, used in the previous version, has inherent uncertainties in its structure due to the utilized decoding scheme. Hence it is challenging to predict and can be attributed to the lower performance of distributed schemes in the previous work.

Comparing with HEVC-NoMotion and MuLE, we can observe the clear downside of using distributed coding schemes. Quantitatively, HEVC-NoMotion and MuLE achieve 4.18 dB and 3.52 dB gain in BD-PSNR, and 66.09% and 57.34% reduction in BD-Rate, respectively, in comparison to our approach. On the other hand, these schemes involve computationally extensive operations and may only be suited for broadcasting applications.

Although the compression performance of the distributed coding paradigm lacks behind the best conventional coding schemes, we emphasize that the application areas and goals are different and we focus on the encoding complexity. Therefore, we discuss the performance of the proposed scheme in comparison to the conventional coding schemes in terms of encoding time. HEVC-Intra does have a complex encoding scheme, even though it does not exploit inter-view redundancy between the views. On the other hand, the other two schemes, HEVC-NoMotion and MuLE, require inter-view communication to exploit the redundancy at the

TABLE 7. Average coding performance in terms of BD-PSNR and BD-Rate compared to HEVC-Intra.

	Proposed		DLFC-I [12]		Checker-MultiSI [13]		HEVC-NoMotion		MuLE [36]	
Sequence	BD-PSNR [dB]	BD-Rate (%)	BD-PSNR [dB]	BD-Rate (%)	BD-PSNR [dB]	BD-Rate (%)	BD-PSNR [dB]	BD-Rate (%)	BD-PSNR [dB]	BD-Rate (%)
Bikes	6.27	−64.8	5.67	−57.8	1.67	−25.7	9.46	−84.2	8.30	−82.9
Danger	5.43	−59.4	4.64	−53.1	1.86	−27.2	8.78	−78.9	9.48	−85.3
Desktop	3.78	−43.9	3.04	−32.4	0.58	−7.3	8.81	−84.8	7.83	−83.4
Flowers	7.21	−71.0	7.24	−67.7	2.35	−32.3	10.98	−86.0	9.69	−86.2
Fountain	5.40	−54.6	3.52	−39.1	1.59	−22.2	10.33	−82.8	8.06	−79.8
Friends	5.34	−62.2	4.75	−54.6	1.19	−21.4	9.91	−90.5	9.52	−90.6
Stone	4.46	−61.4	4.08	−58.3	1.45	−26.9	8.89	−88.1	8.66	−89.2
Vespa	4.56	−55.2	2.85	−37.4	1.42	−23.9	9.01	−85.8	7.03	−81.0
Average	5.31	−59.1	4.47	−50.1	1.51	−23.4	9.52	−85.1	8.57	−84.8

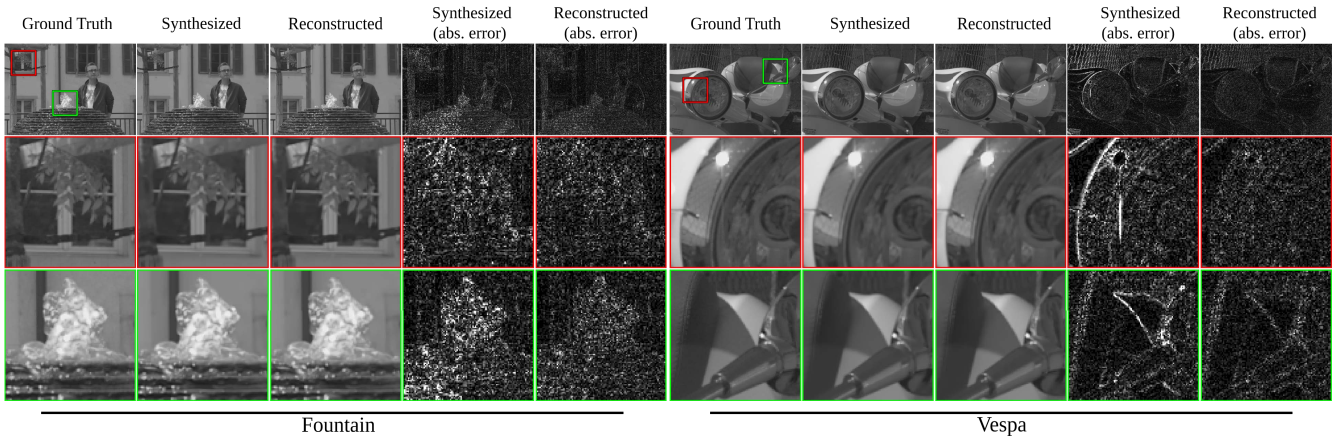


FIGURE 8. Visual comparison between the outputs of stages in the proposed decoding scheme to decode the central view of the two LF sequences, i.e., *Fountain* and *Vespa* at quantization index $M = 8$. The ground truth image and its corresponding zoomed patches are shown on the left. The synthesized and the reconstructed WZ view along with the corresponding absolute errors (range normalized to 0.00 – 0.04) are shown in the next four columns. The zoomed patches are extracted from the highlighted regions in the ground truth images.

encoder, which also results in additional overhead in the encoding time and increases complexity of the encoding architecture. For example, our measurements show that, on average, encoding the light field with the proposed method is 8 to 10 times faster compared to HEVC-Intra depending on the quantization index, whereas it is 12 to 18 times faster than HEVC-NoMotion. In comparison to MuLE, our scheme is 20 to 30 times faster.

It is well-known that distributed coding schemes offer high efficiency encoding by compromising on the simplicity of the decoder [6]. The major contributor in the decoding complexity in our implementation is the iterative LDPCA decoder. Although the iterative LDPCA decoder provides near optimal performance, due to its iterative nature it requires further work on speeding up the iterative decoding for real-time decoding applications. For instance, in the proposed scheme, decoding of a WZ view can be 300 to 1300 times slower than encoding it, depending on the quantization index. Neglecting the fact that the implemented solution for decoding is not optimized, in comparison to HEVC decoder, we have noted that the implementation of the proposed decoding scheme can be approximately 3 orders of magnitude slower.

2) VISUAL ANALYSIS

Fig. 8 illustrates the outputs of the stages in the proposed decoding scheme. In the second column, we can note that the synthesized view provides accurate information about the WZ view in most of the regions. Still, higher errors can be observed in challenging areas such as non-Lambertian surfaces and occluded regions. At the same time, the errors in these areas in the reconstructed views are corrected as observed by the limited error magnitude, which is an outcome of utilizing successfully decoded WZ views for the final reconstruction.

V. CONCLUSION AND FUTURE WORK

We proposed and evaluated deep learning models for distributed light field coding, focusing on the two most critical aspects of side information generation: the prediction and the residual estimation. The proposed models significantly outperform state-of-the-art distributed coding schemes and HEVC-Intra. We have shown that the *Cross* arrangement of reference key views provides higher quality prediction, which improved the overall RD performance compared to the previous approach. Additionally, we propose a deep learning architecture that estimates the residual signal at the coefficient level. We have shown that combining common and specialized filters employed jointly with PDBs allows further performance gains.

We have studied the challenge of distributed coding systems to provide similar performance as offered by the conventional encoding tools while maintaining the low encoding complexity. In future, we aim to further minimize the performance gap between the two coding paradigms by leveraging the latest techniques to model the correlation noise. We further aim to explore light fields with wider baselines, e.g., from large camera arrays, where the constraint on the encoding complexity is more relevant. Additionally, we plan to eliminate the feedback channel requirement by accurately estimating the required number of syndrome bits at the encoder.

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the High Efficiency Video Coding (HEVC) standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[2] F. Pereira, L. Torres, C. Guillemot, T. Ebrahimi, R. Leonardi, and S. Klomp, “Distributed video coding: Selecting the most promising application scenarios,” *Signal Process. Image Commun.*, vol. 23, no. 5, pp. 339–352, 2008.

[3] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1991, p. 409.

[4] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, “The DISCOVER codec: Architecture, techniques and evaluation,” in *Proc. EURASIP PCS*, 2007, pp. 1–4.

[5] J. Ascenso *et al.*, “The VISNET II DVC codec: Architecture, tools and performance,” in *Proc. IEEE ESPC*, 2010, pp. 2161–2165.

[6] B. Girod, A. M. Aaron, S. Rane, and D. Rebollo-Monedero, “Distributed video coding,” *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[7] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive codes for distributed source coding,” *Signal Process.*, vol. 86, no. 11, pp. 3123–3130, 2006.

[8] X. Guo, Y. Lu, F. Wu, W. Gao, and S. Li, “Free viewpoint switching in multi-view video streaming using Wyner-Ziv video coding,” *Proc. SPIE Trans. VCIP*, vol. 6077, 2006, pp. 298–305.

[9] F. Dufaux, W. Gao, S. Tubaro, and A. Vetro, “Distributed video coding: Trends and perspectives,” *EURASIP J. Image Video Process.*, vol. 2009, Apr. 2010, Art. no. 508167.

[10] X. Zhu, A. Aaron, and B. Girod, “Distributed compression for large camera arrays,” in *Proc. IEEE SSP*, 2003, pp. 30–33.

[11] A. Aaron, P. Ramanathan, and B. Girod, “Wyner-Ziv coding of light fields for random access,” in *Proc. IEEE MMSP*, 2004, pp. 323–326.

[12] M. U. Mukati, M. Stepanov, G. Valenzise, F. Dufaux, and S. Forchhammer, “View synthesis-based distributed light field compression,” in *IEEE Proc. ICMEW*, 2020, pp. 1–6.

[13] M. Salmistraro, J. Ascenso, C. Brites, and S. Forchhammer, “A robust fusion method for multiview distributed video coding,” *EURASIP J. Adv. Signal Process.*, vol. 2014, p. 174, Dec. 2014.

[14] H. PhiCong, S. Perry, and X. HoangVan, “Adaptive content frame skipping for Wyner–Ziv-based light field image compression,” *Electronics*, vol. 9, no. 11, p. 1798, 2020.

[15] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, “Learning-based view synthesis for light field cameras,” *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–10, 2016.

[16] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, “Learning to synthesize a 4D RGBD light field from a single image,” in *Proc. IEEE ICCV*, 2017, pp. 2243–2251.

[17] J. Navarro and N. Sabater, “Learning occlusion-aware view synthesis for light fields,” *Pattern Analysis and Applications*. Cham, Switzerland: Springer, 2021, pp. 1–16.

[18] C. Brites and F. Pereira, “Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding,” *IEEE Trans. Circuits Syst. Video Technol*, vol. 18, no. 9, pp. 1177–1190, Sep. 2008.

[19] G. R. Esmaili and P. C. Cosman, “Correlation noise classification based on matching success for transform domain Wyner-Ziv video coding,” in *Proc. IEEE ICASSP*, 2009, pp. 801–804.

[20] X. Huang and S. Forchhammer, “Cross-band noise model refinement for transform domain Wyner-Ziv video coding,” *Signal Process. Image Commun.*, vol. 27, no. 1, pp. 16–30, 2012.

[21] H. V. Luong, L. L. Rakêt, X. Huang, and S. Forchhammer, “Side information and noise learning for distributed video coding using optical flow and clustering,” *IEEE Trans. Image Process.*, vol. 21, no. 12, pp. 4782–4796, Dec. 2012.

[22] I. E. Richardson, *H. 264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. Hoboken, NJ, USA: Wiley, 2004.

[23] W. Ryan, *An Introduction to LDPC Codes*. Boca Raton, FL, USA: CRC Press, 2004.

[24] D. Kubasov, J. Nayak, and C. Guillemot, “Optimal reconstruction in Wyner-Ziv video coding with multiple side information,” in *Proc. IEEE MMSP*, 2007, pp. 183–186.

[25] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assunção, “Data formats for high efficiency coding of Lytro-Illum light fields,” in *Proc. IEEE IPTA*, 2015, pp. 494–497.

[26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. ICLR*, 2015, pp. 1–14.

[27] A. Kendall and Y. Gal, “What uncertainties do we need in Bayesian deep learning for computer vision?” in *Proc. NIPS*, 2017, pp. 5574–5584.

[28] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image style transfer using convolutional neural networks,” in *Proc. IEEE CVPR*, 2016, pp. 2414–2423.

[29] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *Proc. IEEE ECCV*, 2016, pp. 694–711.

[30] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks,” in *Proc. IEEE ICCV*, 2017, pp. 1520–1529.

[31] X. Huang and S. Forchhammer, “Improved side information generation for distributed video coding,” in *Proc. IEEE MMSP*, 2008, pp. 223–228.

[32] “Light field coding common test conditions,” JPEG PLENO, Vancouver, BC, Canada, ISO/IEC JTC 1/SC29/WG1, 2018.

[33] K. Mader, *Lytro-Power-Tools*. 2018. [Online]. Available: https://github.com/kmader/lytro-power-tools

[34] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, calibration and rectification for lenselet-based plenoptic cameras,” in *Proc. IEEE CVPR*, 2013, pp. 1027–1034.

[35] M. Řeřábek and T. Ebrahimi, “New light field image dataset,” in *Proc. IEEE QoMeX*, 2016, pp. 1–2.

[36] P. Schelkens *et al.*, “JPEG Pleno light field coding technologies,” in *Proc. Appl. Digit. Image Process. XLII*, vol. 11137, 2019, Art. no. 111371G.

[37] C. Brites and F. Pereira, “Distributed video coding: Assessing the HEVC upgrade,” *Signal Process. Image Commun.*, vol. 32, pp. 81–105, Mar. 2015.

[38] C. Perra *et al.*, “Performance analysis of JPEG Pleno light field coding,” in *Proc. Appl. Digit. Image Process. XLII*, vol. 11137, 2019, pp. 402–413.

[39] G. Bjøntegaard, “Calculation of average PSNR differences between RD curves,” VCEG, Austin, TX, USA, Rep. VCEG-M33, ITU-T SG16/Q6, 2001.