IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 14, 2019, accepted December 2, 2019, date of publication December 5, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957934

Coding of Light Fields Using Disparity-Based Sparse Prediction

PEKKA ASTOLA, (Member, IEEE), AND IOAN TABUS, (Senior Member, IEEE)
Computing Sciences Unit, Tampere University, 33720 Tampere, Finland

Corresponding author: Pekka Astola (pekka.astola@tuni.fi)

ABSTRACT In this paper we introduce a new light field codec, dubbed WaSPR (warping and sparse prediction on regions), which has additional features and improved performance when compared to our recently introduced WaSP (warping and sparse prediction) codec. We present in detail the overall scheme, including the initial WaSP structure, and the additional new features of WaSPR, including a region based sparse prediction stage, and the addition of a more efficient way to encode the prediction residuals by including separate inter-view coding of residuals obtained at each hierarchical level. The new scheme is demonstrated to improve over both modes of the JPEG Pleno Verification Model software version 2.1, and is shown to provide competitive results when compared against several recently published light field codecs.

INDEX TERMS Light field coding, image coding, plenoptic, multi-view.

I. INTRODUCTION

Light field imaging technologies encompass a wide variety of methods for recording the plenoptic representation of light. The origin of the research into the plenoptic representation of light dates back to the early 20th century [1], [2] with substantial advancement after the introduction of digital imaging [3], [4]. Nowadays, there exist plenoptic cameras for consumer [5] and industrial use [6]. In addition to plenoptic cameras, the research community has been using camera arrays for digital sampling of light fields. Both plenoptic cameras and camera arrays produce data that have characteristics different to those of 2D images due to the higher dimensionality which originates from the recording of the directional information of light rays. The direction of the light rays can be considered as additional sampling in two dimensions resulting in the commonly applied 4D parameterization used for light fields. Compared to traditional 2D imaging technologies, the higher dimensionality of light fields allows for post-processing of light fields into a 3D model of the scene, or into several 2D images each corresponding to a different view point of the scene. For this reason light field imaging has gathered much interest from research and industrial communities.

Our earlier light field compression methods evolved from compression of plenoptic camera images [7] and high-density camera array (HDCA) images [8] into a general light field coding architecture dubbed warping and sparse prediction (WaSP) [9]. Light fields acquired from plenoptic cameras and camera arrays are both interpreted in the 4D prediction framework and the WaSP method is an efficient coding tool for light fields of both types. This property to some extent differentiates WaSP from many other light field coding algorithms, as explained in Section I-C, with several of the algorithms designed to be efficient or practical with only one type of light field data. After being published to the JPEG Pleno Light Field work group in the 80th JPEG meeting in Berlin, WaSP was selected as the verification model (VM) software for VM 1.0 [10], VM 1.1 [11] and as the 4D prediction mode in VM 2.1 [12], all which have been used by the research community for evaluating the performance of light field coding algorithms as will be further discussed in Section I-C. For further discussion of JPEG Pleno light field coding, we refer the reader to [13] and [14].

This paper has two goals: firstly, it presents in detail the architecture and the design of the WaSP light field codec, which is the basis for 4D prediction mode of the JPEG Pleno Light Field draft standard. We introduced WaSP earlier in a conference paper [9], and its publicly available [15] implementation was adopted as the VM 1.0 and 1.1 of JPEG Pleno standardization activity, being also already cited and used as an anchor in the scientific literature. To complete the description of WaSP we present here in more detail the scheme of the initial WaSP codec. Secondly, we introduce a more advanced

The associate editor coordinating the review of this manuscript and approving it for publication was Victor Sanchez.

version of WaSP, called warping and sparse prediction on regions (WaSPR), with inter-view residual coding and more efficient region-based sparse prediction producing state-of-the-art results. Moreover, we show that on densely sampled light fields WaSPR turns out to have better performance compared to the 4D transform mode of the JPEG Pleno Light Field draft standard [14].

The paper is divided as follows. In Section I-A we present the notation and definitions of the light field data as used by the encoder. In Section I-B we describe the most common acquisition methods for the light fields used in this work. In Section I-C we briefly describe the recent light field codecs found in the literature. In Sections II and III the full architecture and warping based view prediction of WaSP and WaSPR are described in detail. In the rest of this paper the intersection of the elements of WaSP and WaSPR is denoted as WaSP(R). In Section IV the new region-based sparse prediction scheme is described. Section V discusses the usage of auxiliary codecs in encoding of texture and normalized disparity [16] data, and presents the inter-view residual coding scheme of WaSPR. In Section VI the hierarchical structure of the codec is detailed together with a discussion of the rate-distortion (R-D) optimization algorithm for the view prediction residuals. In Section VII details on the codestream of the proposed codec are specified. R-D results are provided in Section VIII, and finally conclusions are provided in Section IX.

A. DEFINITIONS AND NOTATIONS

We use the parameterization of the plenoptic function by two planes: the first, having coordinates (t, s), being the angular view point plane, and the second plane having coordinates (v, u) in any angular view as illustrated in Figure 1. The array of angular or sub-aperture views is indexed by the rectangular grid,

$$\begin{aligned} t &\in \{0, \ldots, T-1\}, \\ s &\in \{0, \ldots, S-1\}, \end{aligned} \tag{1}$$

and each sub-aperture view, or sub-aperture image, is defined as a rectangular array of pixels,

$$\begin{aligned} v &\in \{0, \ldots, V-1\}, \\ u &\in \{0, \ldots, U-1\}. \end{aligned} \tag{2}$$

All the texture and disparity data to be encoded are defined over the above two grids, hence being considered 4D data, and we call 4D prediction the operations involving simultaneously the data at various indices along the four coordinates, while 2D prediction (or encoding) refers to operations involving only the coordinates (2).

The texture component of the light field is defined on the above two grids (1) and (2), where the color component c of a single pixel is identified as $X(t, s, v, u, c)$, where $c \in \{0, \ldots, N_c - 1\}$ is the color index with N_c being the number of color components. We refer to the entire texture part light field as $\mathbf{X}$, which is a five-dimensional array, with the generic element $X(t, s, v, u, c)$ having the limits of the indices

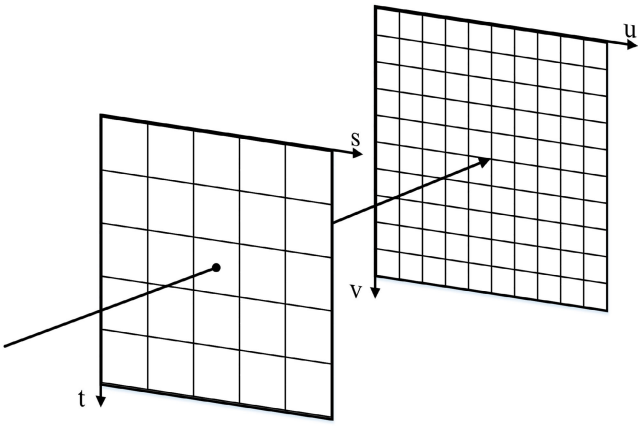

FIGURE 1. In the 4D two-plane parameterization of light fields, each angular view or view point (t, s) will have its corresponding sub-aperture image, sampled on the plane (v, u).

given by (1), (2) and $c \in \{0, \ldots, N_c - 1\}$. We refer to the angular view at coordinates (t, s) as the $V \times U \times N_c$ color image $\mathbf{X}(t, s)$, and hence the data to be encoded is the set of texture views,

$$\{\mathbf{X}(t, s)\}_{0 \le t \le T-1; 0 \le s \le S-1}. \tag{3}$$

The final decoded light field data is identified by the superscript $^{\text{dec}}$ on the corresponding elements of the input texture views, e.g.,

$$\{\mathbf{X}^{\text{dec}}(t, s)\}_{0 \le t \le T-1; 0 \le s \le S-1}.$$

In the process of warping the texture views we use additional input data specifying the normalized disparity, defined as $\tilde{Z}(t, s, v, u)$ on the same two grids (1) and (2). We refer to the normalized disparity angular view at coordinates (t, s) as the $V \times U$ image $\tilde{\mathbf{Z}}(t, s)$. All normalized disparity views are defined as a set,

$$\{\tilde{\mathbf{Z}}(t, s)\}_{0 \le t \le T-1; 0 \le s \le S-1}, \tag{4}$$

from which the encoder uses as input data only a very sparse subset. Since normalized disparity data is needed also at the decoder, we denote $\tilde{\mathbf{Z}}^{\text{dec}}$ the decoded normalized disparity data, which is used for warping operations at both the encoder and the decoder.

During the view prediction stage of a given angular view (t, s), the already processed (encoded and decoded) views are used for prediction. The prediction stage includes a disparity based warping followed by a merging procedure, in which multiple warped views are blend together to form a more complete prediction. We denote the predicted texture view, obtained after the actions of warping and merging, as $\mathbf{X}^{\text{dec}}_{\text{merged}}(t, s)$, and similarly for the normalized disparity view $\tilde{\mathbf{Z}}^{\text{dec}}_{\text{merged}}(t, s)$. Due to occlusions some pixel locations of the merged views may still have an undefined value. The merged views are further subjected to inpainting, in which the relatively sparse set of pixel locations missing in all of the warped views are filled with neighboring values. We denote the inpainted texture views of the light

field as $\mathbf{X}^{\text{dec}}_{\text{inpainted}}(t,s)$ and similarly for normalized disparity $\tilde{\mathbf{Z}}^{\text{dec}}_{\text{inpainted}}(t,s)$.

Each view $\mathbf{X}(t,s)$ of the light field array has an associated hierarchical level $H(t,s) \in \mathbb{N}$ reflecting the order in which the views are predicted one from the others. These levels are marked in a $T \times S$ hierarchy matrix $\mathbf{H}$. The following matrix illustrates a possible view configuration for $T = S = 5$ with highest hierarchical level $h_m = 3$,

$$\mathbf{H} = \begin{bmatrix} 1 & 3 & 2 & 3 & 1 \\ 3 & 2 & 3 & 2 & 3 \\ 2 & 3 & 1 & 3 & 2 \\ 3 & 2 & 3 & 2 & 3 \\ 1 & 3 & 2 & 3 & 1 \end{bmatrix},$$

where the reference views are located at the center and the corners of the view array (where the hierarchical level is $H(t,s) = 1$), and the subsequent hierarchical levels 2 and 3 are occupied by the rest of the views. For further discussion of $\mathbf{H}$ see Section VI.

There are N_{ref} views on the first hierarchical level of the light field and their indices in the angular grid are denoted as (t_i^R, s_i^R), where $i \in \{0, \ldots, N_{ref} - 1\}$. For ease of notations we refer to a view on the lowest hierarchical level by using the index i, instead of using its angular indices. Similarly, the intermediate views in the light field are indexed by (t_j^I, s_j^I), where the linear index $j \in \{0, \ldots, N_{int} - 1\}$ is an alternative way to specify the index of an intermediate view.

Each intermediate view j has its own set of texture reference views with their set of indices denoted as Ω_j^X, and a set of normalized disparity reference views with their set of indices denoted as $\Omega_j^{\tilde{Z}}$. The number of texture reference views for intermediate view j is denoted as $N_j^X = |\Omega_j^X|$, and similarly $N_j^{\tilde{Z}} = |\Omega_j^{\tilde{Z}}|$. Both the texture, and the normalized disparity reference views are obtained from the decoded views $\mathbf{X}^{\text{dec}}$ and $\tilde{\mathbf{Z}}^{\text{dec}}$.

The total number of bits B available for encoding a given light field $\mathbf{X}$ is converted to bits per pixel (bpp) by,

$$\mathcal{R} = \frac{B}{TSVU}, \tag{5}$$

where $\mathcal{R}$ is further divided as,

$$\mathcal{R} = \sum_{i=0}^{N_{ref}-1} \left(\mathcal{R}_i^{ref} + \mathcal{R}_i^{\tilde{Z}} \right) + \sum_{j=0}^{N_{int}-1} \mathcal{R}_j^{res}, \tag{6}$$

with $\{\mathcal{R}_i^{ref}\}$, $\{\mathcal{R}_i^{\tilde{Z}}\}$, and $\{\mathcal{R}_j^{res}\}$, representing the reference view rates, the normalized disparity rates, and the view prediction residual rates respectively. The bit depth of the texture component of the light field $\mathbf{X}$ is b, i.e., $X(t,s,v,u,c) \in \{0, \ldots, 2^b - 1\}$ and bit depth is preserved during encoding and decoding.

B. LIGHT FIELD ACQUISITION

Light fields are usually obtained either using a plenoptic camera or an array of conventional cameras as done in the case of an HDCA. A plenoptic camera, such as the Lytro Illum, see for example the dataset in [17], differs from regular digital cameras by having a microlens array in front of the imaging sensor. The microlens array is a hexagonal grid of small lenses (lenslets) each covering a small fraction of the total number of pixels of the imaging sensor. The pixels under each microlens are illuminated by light rays entering the main lens from different directions of the scene. This allows the microlenses to record directional information of the light rays. The 2D positions on the imaging sensor under each microlens correspond to different view points (t,s), and an individual pixel in each view point is indexed by the microlense index (v,u). The performance of the plenoptic camera is limited by the dimensions of the imaging sensor and the dimensions of the microlenses, and this results in a limited angular and spatial separation between successive view points. However, the plenoptic camera can be implemented in a single device, making it a consumer product comparable to a regular digital camera.

The images acquired by plenoptic cameras are further processed to obtain a rectified array of views [18], which is the plenoptic input data considered in the current paper following the specifications in the JPEG Pleno common test conditions (CTC) [19]. Compared to the plenoptic camera, a HDCA produces light fields with higher image resolution together with wider angular and spatial separation between view points. In a typical HDCA, each view (t,s) is acquired with a high resolution imaging sensor producing a higher definition image when compared to the array of views acquired with a plenoptic camera.

C. LOSSY LIGHT FIELD CODING

Several different approaches for encoding 4D light fields in lossy manner have been proposed in the literature. The existing codecs can be roughly divided into: pseudo-temporal codecs (exploiting existing inter-view redundancies using existing codecs such as HEVC [20]), transform based codecs (directly exploiting the 4D structure of the light field), and predictive codecs (which attempt to exploit directly the 4D redundancies but use predictive transforms instead of a fixed transform). Many of the proposed light field coding algorithms try to make efficient use of standardized 2D image and video codecs in various coding stages. Some of the codecs are designed for dense light fields (see Figure 2) and others for sparsely sampled light fields (see Figure 3) with only few attempting to encode both types of light fields.

The pseudo-temporal encoding approach has been one of the most prominent in the literature [21]–[25]. Such an encoder considers the static light field to be a video sequence obtained using particular scan orders of the views (t,s). The video sequence can be obtained as a sequence of individual sub-aperture views [21], but the approach has also been demonstrated on the raw lenslet image without such preprocessing [22]. State-of-the-art video codecs are highly efficient in exploiting inter-view redundancies and this property can be exploited when considering the sub-aperture views as pseudo-temporal sequence of frames. The rate-distortion

FIGURE 2. A densely sampled light field with small inter-view disparities. The figure illustrates a set of 3 × 3 views from light field Greek with high similarity between the sub-aperture images due to the closeness of the view points (t, s).

FIGURE 3. A sparsely sampled light field with large inter-view disparities. The figure shows a set of 3 × 3 views from light field Set2, illustrating that large distances between the sampled view points in the (t, s) plane imply significant differences of the corresponding sub-aperture images. For this type of light fields, pixel-wise alignment using warping is necessary to efficiently exploit the inter-view redundancy between the sub-aperture images.

performance of different pseudo-temporal sequence methods depends on the scan order of the view points, the choice of the video codec, and on the tuning of the codecs parameters. In [23] the light field acquired using a plenoptic camera is interpreted as a multi-view sequence and the multi-view extension of HEVC is used in a novel rate allocation scheme. In [24] the sub-aperture views of the light field are divided into quadrants with each quadrant exploiting both the center view, and the neighboring views, within an optimized scan order. An approach to maximize the inter-view redundancy on a plenoptic image by designing a suitable coding order of the views is presented for HEVC in [25].

In [26] a 4D discrete cosine transform (DCT) based codec for plenoptic camera images is proposed for coding of dense light fields. Reminding the 2D DCT scheme of the legacy JPEG codec, the 4D DCT approach divides the sub-aperture images of the lenslet image into 4D blocks, and each block is processed by applying 4D DCT, then the significant transform coefficients are specified by a hexadeca-tree, and finally the significant coefficients are entropy coded. The method is called multidimensional light field encoder using 4D transforms and hexadeca-trees (MuLE) and serves as the 4D transform mode in JPEG Pleno Light Field VM 2.1. A joint homography and low-rank approximation scheme (HLRA) is used in [27] to first align the sub-aperture views and subsequently to obtain a sparse representation of the aligned structure followed by the use of HEVC for encoding the novel representation. HLRA is shown to improve over HEVC pseudo-temporal sequence coding of dense light fields. In [28] a small number of sub-aperture images are encoded using HEVC, and subsequent views are reconstructed using a shearlet transform based prediction scheme. In [29] the plenoptic image is encoded by a sparse set and disparities. The sparse set and the disparities are encoded with HEVC and the rest of the plenoptic image is synthesized using disparity based reconstruction, interpolation, and inpainting. In [30] a motion compensated wavelet lifting scheme is used to encode the sub-aperture images. Similar to our encoding strategy, the disparity map is first estimated and subsequently encoded in the codestream. An elaborated disparity model is used in [16], together with hierarchical disparity compensated inter-view transform, followed by wavelet decomposition and coding using EBCOT [31]. The scheme is demonstrated in encoding of high resolution HDCA data with R-D performance superior to HEVC.

Our earlier lossy light field codecs are all based on disparity compensated predictive coding. In [7] the lenslet image was sliced into a set of non-rectified sub-aperture images from which a set of reference views were selected. The reference views were jointly encoded as a subsampled lenslet image instead of separated images. A segmented version of the scene depth map at the central sub-aperture view was encoded and the displacements of a region from central to all other views were obtained and encoded. For each region in each view a sparse predictor is designed based on its neighboring views.

For sparsely sampled light fields with large inter-view disparities, such as the HDCA images, obtaining a consistent disparity based segmentation over the light field becomes difficult due to significant occlusions. Additionally, the large disparities between neighboring views render the sparse prediction scheme of [7] computationally complex. In [8] the segmentation based sparse prediction approach was replaced by optimal prediction based on a fixed set of reference views with inter-view correspondences obtained from disparity based warping. Each warped reference view provides a prediction of the target view, and the final prediction was obtained using occlusion based segmentation with a separate least squares predictor for each region in each view. In [9]

the prediction scheme of [8] was improved by introducing a hierarchical prediction scheme with residual coding and the overall method was dubbed WaSP.

Recently several papers [32]–[36] have been published with comparisons to JPEG Pleno VM implementations of WaSP for encoding densely sampled light fields such as the ones obtained with a plenoptic camera. These light fields represent a subset of the JPEG Pleno datasets [19]. Graph learning technique is used in [32] at the encoder to capture the inter-view redundancy of the light field and the resulting graph is transmitted losslessly. A subset of views are encoded using HEVC and the remaining views are reconstructed by solving a minimization problem at the decoder side. The results demonstrate performance better than VM 1.0. In [33] a block-based least squares sense optimal linear predictor is used to predict the light field from a subset of HEVC encoded reference views with residual encoded using a low-rank approximation strategy. The scheme is demonstrated to perform better than both of the encoding modes of VM 2.1 on dense light fields. An iterative segmentation, known as a collection of super-rays, is proposed in [34]. A low rank approximation using singular value decomposition (SVD) is then performed on the super-rays and the resulting eigen images are entropy coded using HEVC. The method is demonstrated for densely sampled light fields and shows performance gains when compared to VM 1.1. The super-ray concept was further used together with a reversible graph transform in [35] where the geometry-aware graph based transform (GBT) is used instead of the SVD to sparsify the inter-view redundancies between the super-rays, and the resulting graph transform coefficients together with the disparity information are entropy coded using arithmetic coding. For encoding densely sampled light fields GBT was shown to offer coding performance exceeding VM 1.1 at higher rates. In [36] a hierarchical coding strategy based on Fourier disparity layer (FDL) representation is proposed. A subset of views are encoded using HEVC from which the FDL representation is obtained. The FDL is then used to hierarchically predict subsequent views and view prediction residual is again encoded using HEVC. The method was shown to offer improvements over VM 1.1.

In [37] the performance of the WaSP codec was improved by utilizing a breakpoint adaptive discrete wavelet transform (BPA-DWT) in coding of the normalized disparity views. For encoding HDCA datasets, the BPA-DWT approach was shown to preserve better the discontinuities appearing in the normalized disparity images. Therefore, the warping and merging operations of WaSP can utilize the normalized disparity information more precisely and improvements in the R-D performance were demonstrated. The near-lossless performance of WaSP for encoding of medical light fields was reported in [38] where it was compared against the near lossless performance of the 4D DCT transform [26] and the recent lossless light field codec known as minimum rate predictors [39].

II. ENCODER ARCHITECTURE

In this section the main parts of the WaSP(R) encoder are introduced. The light field encoder takes as input a subset of the texture views (3) and a subset of normalized disparity views (4). These input views are *reference views*, or views at the lowest hierarchical level. The task of the encoder is to create a bitstream from which the decoder can reconstruct all the texture views (3). According to the CTC, we consider experiments where all texture views (3) are available, and only a small subset of normalized disparity views is available. We note an important additional functionality: the decoder can decode from the bitstream the complete set of normalized disparity views (4), which are lossy reconstructions obtained in identical way at the encoder and decoder.

The encoder block scheme is shown in Figure 4. The encoder's main task is to encode the texture part of the light field with the lowest possible distortion, given a specific rate $\mathcal{R}$. In the proposed codec this is achieved by utilizing the normalized disparity information during the prediction of the *intermediate views*. The block scheme for the intermediate view reconstruction is shown in Figure 5. The intermediate view reconstruction stage is a combination of hierarchical texture view prediction and residual coding. Existing image coding tools, in this work labeled as *auxiliary codecs*, such as JPEG 2000 [40] or HEVC, are used for encoding: 1) full texture views at lowest hierarchical level, 2) texture prediction residual for intermediate views, and 3) normalized disparity data at the lowest hierarchical level. In this paper we use JPEG 2000 for encoding the normalized disparity views and HEVC for encoding the texture and residual views.

A. TEXTURE

We encode RGB or YCbCr views, having $N_c = 3$, with V and U being anywhere from 512 for low resolution images, to 2000 for high resolution datasets. In the experimental section results are additionally demonstrated on encoding only the luma (Y) channel for being able to report comparisons with recent light field codecs for which the experimental results were available for luma channel only. In all our experiments, the bit depth for the texture components is $b = 10$.

B. NORMALIZED DISPARITY

For each view (t, s) we define the normalized disparity map at pixel (v, u) as $\tilde{Z}(t, s, v, u)$. The normalized disparity map will be used for creating horizontal and vertical disparity maps between a reference view (t_1, s_1) and any target view (t_2, s_2), which are needed for warping the view (t_1, s_1) to the location of view (t_2, s_2). Denoting the center coordinates for reference and target as $C_x(t_1, s_1)$ and $C_x(t_2, s_2)$ respectively, we obtain the horizontal baseline as,

$$\Delta x = C_x(t_1, s_1) - C_x(t_2, s_2),$$

and similarly for the vertical baseline,

$$\Delta y = C_y(t_1, s_1) - C_y(t_2, s_2).$$

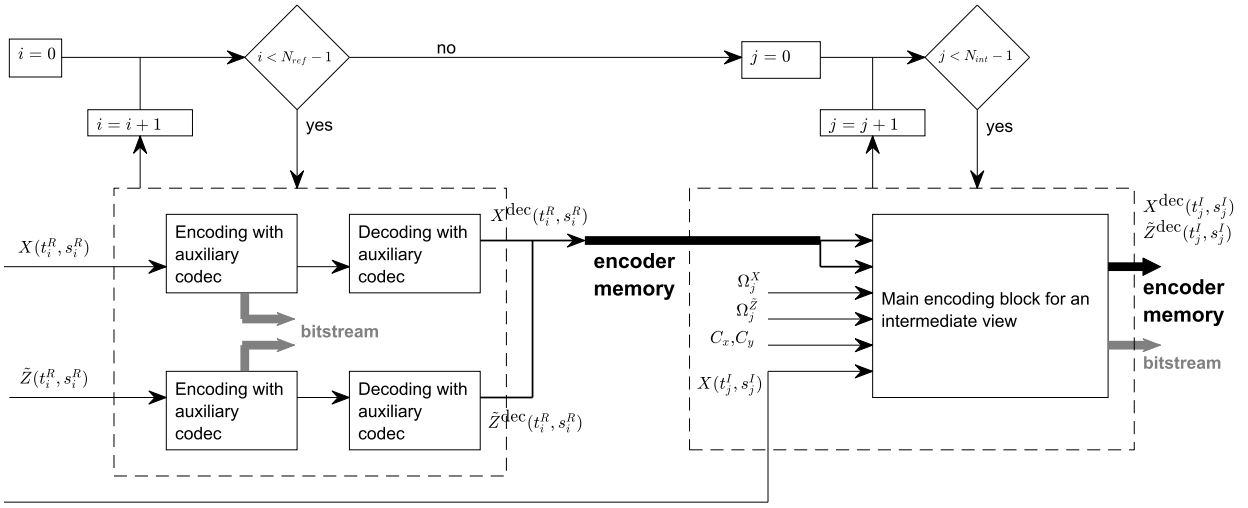

FIGURE 4. WaSP(R) block scheme. The views on the lowest hierarchical level are encoded first. The remaining views of the light field are encoded in the hierarchical order and the block scheme for encoding the intermediate views is depicted in Figure 5. The decoded reference texture and normalized disparity views enter the encoder memory from which they are accessed in the view prediction blocks in Figure 5.

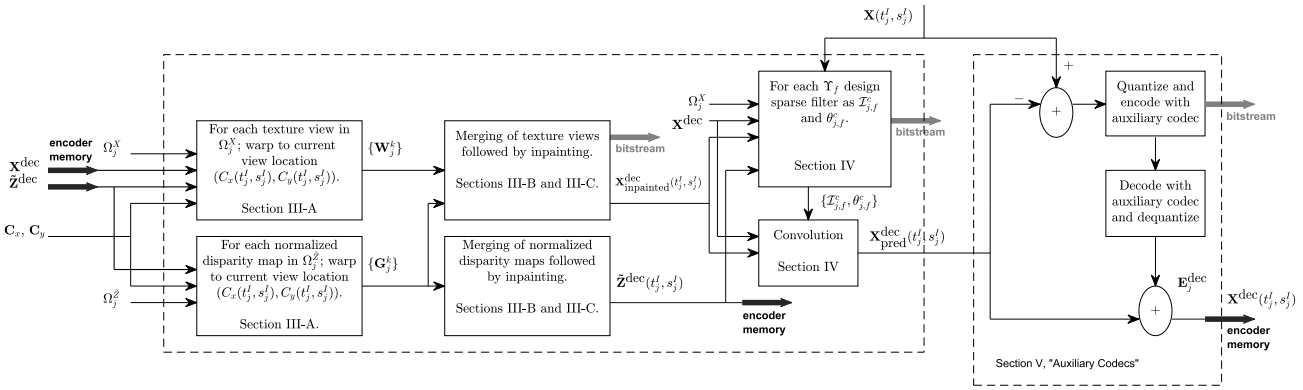

FIGURE 5. Main prediction block of WaSPR for intermediate view (t_j^I, s_j^I), $j \in \{0, \ldots, N_{int-1}\}$ together with prediction residual coding using the auxiliary codec. This diagram is a detailed version of the main encoding block illustrated by a single box in Figure 4.

On the discrete grid, in case of no occlusion, the pixel (v, u) in view (t_1, s_1) corresponds to the pixel $(\hat{v}, \hat{u})$ in view (t_2, s_2) obeying the disparities,

$$\hat{v} = v + \lfloor \tilde{Z}(t_1, s_1, v, u)\Delta y \rceil,$$
$$\hat{u} = u + \lfloor \tilde{Z}(t_1, s_1, v, u)\Delta x \rceil. \quad (7)$$

The rounding to the closest integer operation, denoted $\lfloor \cdot \rceil$ in (7), produces the necessary equations for warping with an integer number of pixels. The subsequent optimal texture prediction operations are introducing further adjustments of the integer precision warping, resulting in an equivalent overall warping operation using fractional pixels.

C. TEXTURE RECONSTRUCTION BASED ON WARPED REFERENCE VIEWS

At each intermediate view $\mathbf{X}(t_j^I, s_j^I)$ the texture view reconstruction includes the following operations:

1) For $k \in \{0, \ldots, N_j^X - 1\}$, pick the k'th view index (t, s) from Ω_j^X. Since all angular indices in Ω_j^X belong to lower hierarchical levels, (t, s) is the angular index of an already encoded and decoded view. Warp $\mathbf{X}^{\text{dec}}(t, s)$ to obtain the warped view $\mathbf{W}_j^k$. Proceed and in the end obtain all warped views $\{\mathbf{W}_j^0, \ldots, \mathbf{W}_j^{N_j^X-1}\}$.
2) Merge the warped views $\{\mathbf{W}_j^0, \ldots, \mathbf{W}_j^{N_j^X-1}\}$ and obtain $\mathbf{X}^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I)$ (see Section III-B).
3) Inpaint $\mathbf{X}^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I)$ to obtain $\mathbf{X}^{\text{dec}}_{\text{inpainted}}(t_j^I, s_j^I)$ (see Section III-C).
4) Perform sparse filtering to obtain $\mathbf{X}^{\text{dec}}_{\text{pred}}(t_j^I, s_j^I)$ (see Section IV).
5) Obtain view prediction residual $\mathbf{E}_j = \mathbf{X}(t_j^I, s_j^I) - \mathbf{X}^{\text{dec}}_{\text{pred}}(t_j^I, s_j^I)$.
6) Encode $\mathbf{E}_j$ using rate $\mathcal{R}_j^{res}$; then decode, and store the result in $\mathbf{E}_j^{\text{dec}}$.
7) Add the decoded view prediction residual to the prediction and obtain $\mathbf{X}^{\text{dec}}(t_j^I, s_j^I)$ which is the final reconstruction of the texture.

The warping operation applies a pixelwise horizontal and vertical displacement on the pixels of the reference view, and the magnitude of the displacement is obtained as a

FIGURE 6. Warping a reference view to a nearby camera position yields almost complete reconstruction. Black regions correspond to scene points which were not seen by the camera at the reference view location. The above image is the result of warping the reference view X(0, 2) to the camera position of the target view X(4, 2).

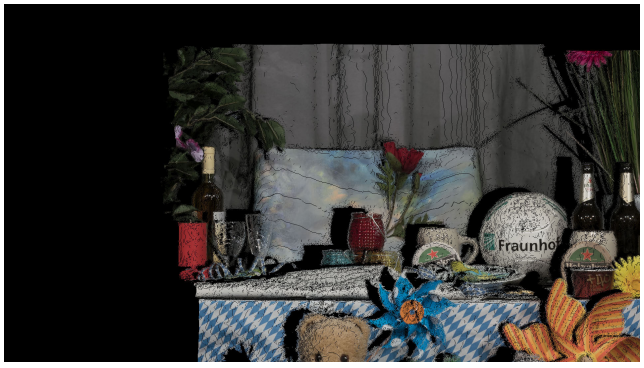

FIGURE 7. Warping a reference view to a faraway camera position produces only a partial reconstruction. Similar to Figure 6 black regions correspond to pixels which are not seen by the camera at the reference view location. The above image is the result of warping the reference view X(20, 98) to the camera position of the target view X(4, 2). Since the reference view resides on the lower right corner of the camera array, the upper and left parts of the view at the target location cannot be reconstructed using warping of the reference view alone.

multiplication of the normalized disparity with horizontal and vertical baselines. The warping algorithm is presented in Section III-A. Step 2 applies the mixing, or merging of the multiple warped reference views into a single predicted intermediate view.

The warping operation in Step 1 may introduce areas of missing pixels which originate from the occluded areas appearing when changing the view point, as shown in Figures 6-7. Warping a view to nearby camera position can be used to construct an almost complete prediction of the view, as shown in Figure 6. However, for views far apart in the angular grid, as seen in Figure 7, warping will introduce large missing areas of pixels due to occlusions. Increasing the number of neighboring views (see hierarchical coding of views in Section VI) usually solves the occlusion problem, leaving only a small number of undefined pixels in the merged image. In Step 3 to complete the predicted view the inpainting algorithm of Section III-C is applied on the merged result. Step 4 applies adjustment of the predicted view using sparse filtering, see Section IV. Steps 5-7 obtain, encode, and decode the prediction residual and obtain also the final decoded view, needed also at the encoder in the case the current view will be used as a reference for encoding other views. The encoding of the prediction residual is optional and when targeting extremely low rates the encoding of prediction residual is usually disabled.

D. NORMALIZED DISPARITY RECONSTRUCTION BASED ON WARPED NORMALIZED DISPARITY VIEWS

Normalized disparity views are predicted similarly to the prediction of texture views. At each intermediate view (t_j^I, s_j^I) the reconstruction of the normalized disparity view $\tilde{\mathbf{Z}}^{\text{dec}}(t_j^I, s_j^I)$ includes the following operations:

1) For $k \in \{0, \ldots, N_j^{\tilde{Z}} - 1\}$ pick the k'th view index (t, s) from $\Omega_j^{\tilde{Z}}$ and warp $\tilde{\mathbf{Z}}^{\text{dec}}(t, s)$ to obtain the warped view $\mathbf{G}_j^k$. Proceed and in the end obtain all warped views $\{\mathbf{G}_j^0, \ldots, \mathbf{G}_j^{N_j^{\tilde{Z}}-1}\}$.
2) Merge the warped normalized disparity views $\{\mathbf{G}_j^0, \ldots, \mathbf{G}_j^{N_j^{\tilde{Z}}-1}\}$ to obtain $\tilde{\mathbf{Z}}^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I)$.
3) Inpaint $\tilde{\mathbf{Z}}^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I)$ to obtain $\tilde{\mathbf{Z}}^{\text{dec}}(t_j^I, s_j^I)$.

The normalized disparity views make no use of the prediction residual since often the normalized disparity views are obtained for a very sparse subset of views and in some cases the encoder receives only one normalized disparity view. Therefore it is not possible to obtain prediction residual for most of the normalized disparity views. For the same reason the normalized disparity views are merged using the median operator instead of the predictive merging algorithm.

III. VIEW PREDICTION

Section II-C already provided an overview of the view prediction algorithm used in WaSP(R). In this section further details are provided on view warping and on the occlusion aware predictive view merging algorithm.

A. VIEW WARPING

View warping together with the view merging algorithm of Section III-B are the first layer of the texture prediction tools, and they transforms the texture and normalized disparity components of the view (t_1, s_1), located at the camera coordinates $C_y(t_1, s_1)$, $C_x(t_1, s_1)$, to the camera coordinates $C_y(t_2, s_2)$, $C_x(t_2, s_2)$ of the view (t_2, s_2). Therefore, view warping produces a prediction of view (t_2, s_2) conditional on texture and normalized disparity at view (t_1, s_1). The quality of the prediction is constrained by both the quality of the normalized disparity at view (t_1, s_1), the precision of the camera center coordinates $\mathbf{C_y}$, and $\mathbf{C_x}$, and the quality of the reference view texture.

The warping procedure described in Algorithm 1 begins with the initialization of the warped normalized disparity and the warped texture views at lines 4-9. The pixel values in

the warped normalized disparity view are initialized to minus infinity, which enables for quick detection of remaining undefined pixels after warping. At lines 14-15, for each pixel (v, u), the horizontal and vertical disparities are obtained by multiplying the normalized disparity $\tilde{Z}(t_1, s_1, v, u)$ with the vertical and horizontal baselines Δy, Δx which have been obtained from the camera center coordinate arrays $\mathbf{C_y}$ and $\mathbf{C_x}$ at lines 10-11. The obtained disparities D_v, D_u are rounded to the nearest integer. The current pixel coordinates (v, u) at view (t_1, s_1) are transformed to the warped pixel coordinates $(\hat{v}, \hat{u})$ at view (t_2, s_2) on lines 16-17.

The warped pixel coordinates may reside outside the image dimensions and an out of bounds test is performed at line 18. If the warped pixel coordinates lie inside the image dimensions the algorithm proceeds to check for a possible occlusion. The occlusion check at line 20 compares the normalized disparity value at the warped pixel coordinates $(v+D_v, u+D_u)$ in the warped normalized disparity view to the normalized disparity value (v, u) at the reference, and overwrites the warped normalized disparity value only if it is smaller than the one at the reference. This ensures that pixels from objects closer to the camera plane are not overwritten by more distant objects. If the warped pixel coordinates pass the occlusion test, the warping action takes place at lines 21-25 and the normalized disparity and texture values of view (t_1, s_1) at pixel coordinates (v, u) are copied to the warped normalized disparity and texture views at (t_2, s_2) in pixel coordinates $(v + D_v, u + D_u)$.

B. TEXTURE VIEW MERGING USING OCCLUSION BASED SPARSE LINEAR PREDICTION

In this section we present the linear prediction model used in WaSP(R) for texture view merging. First we describe the sets of reference texture and normalized disparity views used in the view merging operations, followed by the introduction of the occlusion based segmentation used in the linear prediction stage. Then the method for obtaining the linear prediction coefficients is described in detail.

Let us consider an intermediate view (t_j^I, s_j^I) with N_j^x reference texture views indexed by the elements of $\Omega_j^X = \{(t_0^X, s_0^X), \ldots, (t_{N_j^x-1}^X, s_{N_j^x-1}^X)\}$. The warped reference texture views are denoted as,

$$\begin{aligned}\mathbf{W}_j^0 &= \mathbf{X}^{\mathrm{dec}}{}_W^{(t_0^X, s_0^X)}(t_j^I, s_j^I),\\ &\vdots\\ \mathbf{W}_j^{N_j^x-1} &= \mathbf{X}^{\mathrm{dec}}{}_W^{(t_{N_j^x-1}^X, s_{N_j^x-1}^X)}(t_j^I, s_j^I),\end{aligned}$$

and the warped normalized disparity views are denoted as,

$$\begin{aligned}\mathbf{G}_j^0 &= \tilde{\mathbf{Z}}^{\mathrm{dec}}{}_W^{(t_0^X, s_0^X)}(t_j^I, s_j^I),\\ &\vdots\\ \mathbf{G}_j^{N_j^x-1} &= \tilde{\mathbf{Z}}^{\mathrm{dec}}{}_W^{(t_{N_j^x-1}^X, s_{N_j^x-1}^X)}(t_j^I, s_j^I).\end{aligned}$$

Algorithm 1 View Warping of View at Camera Location of the View (t_1, s_1) to Camera Location of the View (t_2, s_2)

1: **procedure** ViewWarping($\tilde{\mathbf{Z}}^{\mathrm{dec}}$, $\mathbf{X}^{\mathrm{dec}}$, $\mathbf{C_y}$, $\mathbf{C_x}$, t_1, s_1, t_2, s_2)
2: $\mathcal{V} = \{0, \ldots, V-1\}$
3: $\mathcal{U} = \{0, \ldots, U-1\}$
4: **for all** v, u **do**
5: $\tilde{Z}_W^{(t_1,s_1)}(t_2, s_2, v, u) = -\infty$
6: **for all** c **do**
7: $\tilde{X}_W^{(t_1,s_1)}(t_2, s_2, v, u, c) = 0$
8: **end for**
9: **end for**
10: $\Delta y = C_y(t_1, s_1) - C_y(t_2, s_2)$
11: $\Delta x = C_x(t_1, s_1) - C_x(t_2, s_2)$
12: **for all** $v \in \mathcal{V}$ **do**
13: **for all** $u \in \mathcal{U}$ **do**
14: $D_v = \lfloor \tilde{Z}^{\mathrm{dec}}(t_1, s_1, v, u)\Delta y \rceil$
15: $D_u = \lfloor \tilde{Z}^{\mathrm{dec}}(t_1, s_1, v, u)\Delta x \rceil$
16: $\hat{v} = v + D_v$
17: $\hat{u} = u + D_u$
18: **if** $\hat{v} \in \mathcal{V} \wedge \hat{u} \in \mathcal{U}$ **then**
19: $\hat{Z} = \tilde{Z}^{DEC}(t_1, s_1, v, u)$
20: **if** $\tilde{Z}_W^{(t_1,s_1)}(t_2, s_2, \hat{v}, \hat{u}) < \hat{Z}$ **then**
21: $\tilde{Z}_W^{(t_1,s_1)}(t_2, s_2, \hat{v}, \hat{u}) = \hat{Z}$
22: **for all** c **do**
23: $\hat{X} = X^{DEC}(t_1, s_1, v, u, c)$
24: $\tilde{X}_W^{(t_1,s_1)}(t_2, s_2, \hat{v}, \hat{u}, c) = \hat{X}$
25: **end for**
26: **end if**
27: **end if**
28: **end for**
29: **end for**
30: **end procedure**

The goal of the encoder is to synthesize, or predict, the texture part of the intermediate view (t_j^I, s_j^I) using the information available in both sets $\{\mathbf{W}_j^0, \ldots, \mathbf{W}_j^{N_j^x-1}\}$ and $\{\mathbf{G}_j^0, \ldots, \mathbf{G}_j^{N_j^x-1}\}$. In the following section we introduce an occlusion based segmentation used in a sparse subset selection on the warped pixel values so that only the most relevant parts of the warped reference views are utilized for intermediate view prediction.

1) OCCLUSION CLASSES

For any warped view, due to occlusions, it is possible that not all pixel locations get assigned a value in the main loop of the warping algorithm of Algorithm 1. Different reference views produce different occlusion patterns, and the aggregation of this information can be used to infer a useful segmentation of the predicted intermediate view. Each occlusion class corresponds to a different sparse subset of the reference views used to obtain the best linear predictor involving only the relevant reference views at each pixel location. We introduce the

non-occlusion operator $\delta(G(v,u))$ as,

$$\delta(G(v,u)) = \begin{cases} 1, & G(v,u) > -\infty \text{ (non-occluded)} \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

allowing to define the vector-state of occlusions at the pixel (v,u), as follows. The occlusion state at a pixel (v,u) in the current view (t_j^I, s_j^I) is a binary vector $\mathbf{b}_j^{(v,u)}$ of length N_j^x having the elements,

$$b_j^{(v,u)}(k) = \delta(G_j^k(v,u)), \quad k \in \{0,\ldots,N_j^x - 1\}.$$

We introduce the matrix $\mathbf{F}_j$ of dimensions $V \times U$ defined by,

$$F_j(v,u) = \sum_{k=0}^{N_j^x-1} 2^k b_j^{(v,u)}(k),$$

where each element $F_j(v,u)$ is a label containing the integer representation of the binary vector $\mathbf{b}_j^{(v,u)}$. The class $F_j(v,u) = 0$ corresponds to the pixels where all reference views are occluded, and therefore is not used for prediction. In essence $\mathbf{F}_j$ is a segmentation with $2^{N_j^x}$ classes of the pixels in the warped images $\mathbf{G}_j^k$ and $\mathbf{W}_j^k$. The occlusion classes for intermediate view j are given by the set $\mathcal{C}_j = \{0,\ldots,2^{N_j^x}-1\}$. In the following section we describe how to design a set of merging weights for each of the classes in $\mathcal{C}$.

2) DESIGN OF THE LINEAR PREDICTORS

For a given color component c, the merging weight matrix, or coefficient matrix Θ_j^c has dimensions $2^{N_j^x} \times N_j^x$ and contains the weights used to obtain the merged intermediate texture view $\mathbf{X}^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I)$. The merging weights are given for each occlusion class $m \in \mathcal{C}_j$ as the m'th row in Θ_j^c. The weight of the n'th reference view is given by the n'th column of Θ_j^c. For a pixel (v,u) belonging to class m we have $\mathbf{F}_j(v,u) = m$ and the merged intermediate texture view is evaluated as,

$$X_{model}(t_j^I, s_j^I, v, u, c) = \sum_{k=0}^{N_j^x-1} W_j^k(v,u,c)\delta(G_j^k(v,u))\Theta_j^c(m,k). \quad (9)$$

The matrix of optimal parameters Θ_j^c is obtained for each of its rows by performing a least-squares design for minimizing the sum of squared residuals $\sum_{(v,u)}(\epsilon(v,u,c))^2$ for all (v,u) so that $\mathbf{F}_j(v,u) = m$, using the linear model (9),

$$X(t_j^I, s_j^I, v, u, c) = X_{model}(t_j^I, s_j^I, v, u, c) + \epsilon(v,u,c),$$

where $\epsilon(v,u,c)$ becomes a function of the parameters Θ_j^c through (9). Hence, the full matrix Θ_j^c is obtained by solving $2^{N_j^x}$ least squares problems. The coefficients are further rounded to 10 bits in the fractional part, obtaining the index of quantization,

$$\overline{\Theta}_j^c = \lfloor \Theta_j^c 2^{10} \rceil. \quad (10)$$

Both the encoder and decoder will use the quantized version $\overline{\Theta}_j^c/2^{10}$ instead of the full precision Θ_j^c and the indices of quantization are written in raw format in the bitstream.

The predicted intermediate view is defined as the linear combination (9) of the reference views over the occlusions classes $m \in \mathcal{C}_j$ using the quantized parameters from (10) and the model output is further rounded to the nearest integer, to obey the constraint on the integer alphabet of the decoded image, resulting in,

$$X^{\text{dec}}_{\text{merged}}(t_j^I, s_j^I, v, u, c) = \lfloor X_{model}(t_j^I, s_j^I, v, u, c) \rceil,$$

with obtained values further clipped to the known range of pixel values $\{0,\ldots,2^b - 1\}$.

C. INPAINTING

During the warping process some of the pixels $(\hat{v}, \hat{u})$ remain undefined due to occlusions. These locations can be detected using the non-occlusion operator (8) after the execution of Algorithm 1. When predicting an intermediate view with several reference views the frequency of occluded pixels is low and their effect on the image can be approximated by salt-and-pepper noise. Based on this observation, in WaSP(R) we apply a simple median filtering based inpainting procedure to fill in the undefined values prior to sparse filtering and residual coding. The inpainting procedure sequentially scans the missing values of the image in row-wise scan order, each time filling them by the median of their (non-occluded) 3×3 neighborhood. Each successfully filled (i.e., filtered) pixel is then considered as non-occluded and the algorithm quickly fills the occluded parts of the image with the procedure repeated until all missing pixel locations are filled. Same inpainting algorithm is used for both texture and normalized disparity.

IV. REGION-BASED SPARSE PREDICTION

In WaSPR the region-based sparse filter is used to perform the final adjustment of the merged and inpainted intermediate view $\mathbf{X}^{\text{dec}}_{\text{inpainted}}(t_j^I, s_j^I)$, resulting in an equivalent refinement of the initial integer-pixels warping to a fractional pixel precision displacements. The sparse filter reduces the magnitude of the prediction residual by adjusting the texture component in the N_r different disparity regions defined as a set of pixel locations,

$$\Upsilon_f = \{(v,u)\}, f \in \{0,\ldots,N_r - 1\},$$

with the sparse linear predictors obtained using a greedy sparse algorithm. Next we will describe how the regions and their corresponding sparse filters are obtained.

A. OBTAINING DISPARITY BASED REGIONS

The regions Υ_f, for which the sparse predictors are designed, are obtained from the normalized disparity map $\tilde{\mathbf{Z}}^{\text{dec}}(t_j^I, s_j^I)$. The segmentation process is recursive, each time subdividing a given region into two new regions, with the threshold being the median value of the given region. For adequately small

N_r this simple segmentation process obtains a segmentation with rather uniform sizes $|\Upsilon_f|$, and for example with $N_r = 2$ the process obtains a foreground-background segmentation which we found very useful when using the proposed scheme for coding of plenoptic camera images at very low rates.

B. DESIGN OF THE OPTIMAL SPARSE PREDICTORS FOR EACH REGION

Let us denote by $\mathbf{X}^c(t_j^I, s_j^I)$ the $(V \times U)$ matrix that is the c'th color component of the texture image $\mathbf{X}(t_j^I, s_j^I)$. We want a model for $\mathbf{X}^c(t_j^I, s_j^I)$, and we linearize the matrix by columns, obtaining the $(VU \times 1)$-vectors $\bar{\mathbf{y}}_j^c$, $\forall c \in \{0, \ldots, N_c - 1\}$. For each region Υ_f we select the corresponding rows as $(|\Upsilon_f| \times 1)$-vectors $\mathbf{y}_{j,f}^c$. We express the linear prediction model to be estimated from the set of matrix equations,

$$\mathbf{y}_{j,f}^c = \mathbf{D}_{j,f}^c \Theta_{j,f}^c + \boldsymbol{\epsilon}_{j,f}^c, \tag{11}$$

where,

$$\begin{aligned}
\mathbf{y}_{j,f}^c &\in \mathbb{R}^{|\Upsilon_f|}, \\
\mathbf{D}_{j,f}^c &\in \mathbb{R}^{|\Upsilon_f| \times ((N_j^X+1)N_{full}+1)}, \\
\Theta_{j,f}^c &\in \mathbb{R}^{(N_j^X+1)N_{full}+1}, \\
\boldsymbol{\epsilon}_{j,f}^c &\in \mathbb{R}^{|\Upsilon_f|},
\end{aligned}$$

where $N_{full} = (2L_j+1)^2$ represents the number of elements in the spatial template $\Psi_{(v,u)}$ defined as the vector of coordinate pairs,

$$\begin{aligned}
\Psi_{(v,u)} = [&(v - L_j, u - L_j), \\
&(v - L_j, u - L_j + 1), \ldots, (v + L_j, u + L_j)],
\end{aligned}$$

and L_j is a configurable parameter for the filter usually in the range of $\{1, 2, 3\}$. The p'th row of the regressor matrix $\mathbf{D}_{j,f}^c$ is constructed in the following way,

1) p'th row of $\mathbf{D}_{j,f}^c$ corresponds to a value at pixel location $(v, u) \in \Upsilon_f$,
2) the columns k, $k \in \{0, \ldots, N_{full} - 1\}$ of the p'th row in $\mathbf{D}_{j,f}^c$ are defined as a specific neighbor in the regressor template $\Psi_{(v,u)}$ for the intermediate view j,

$$D_{j,f}^c(p, k) = X^{\text{dec}}(t_j, s_j, \hat{v}, \hat{u}, c),$$

3) and the rest of the columns $(n + 1)N_{full} + k$ of the p'th row in $\mathbf{D}_{j,f}^c$ are defined as a specific neighbor in the regressor template $\Psi_{(v,u)}$ for reference view $n \in \{0, \ldots, N_j^X - 1\}$,

$$D_{j,f}^c(p, (n+1)N_{full} + k) = X^{\text{dec}}(t_n^X, s_n^X, \hat{v}, \hat{u}, c),$$

where $(\hat{v}, \hat{u}) = \Psi_{(v,u)}(k)$ and $(t_n^X, s_n^X) \in \Omega_j^X$.

The last column of $\mathbf{D}_{j,f}^c$ is set to ones and corresponds to the bias term used in the model. The N_c least squares problems for the model (11) are $\min_{\Theta_{j,f}^c} \|\boldsymbol{\epsilon}_{j,f}^c\|^2$, each having the full model solution,

$$\left(\mathbf{D}_{j,f}^{c\,T} \mathbf{D}_{j,f}^c\right) \Theta_{j,f}^c = \mathbf{D}_{j,f}^{c\,T} \mathbf{y}_{j,f}^c.$$

The task of the sparse predictor is to select the sets $\mathcal{I}_{j,f}^c$ each with only m_j most relevant columns, where the value of m_j is a configuration parameter common to all color components and is usually in the range of $\{5, \ldots, 25\}$. The sparse supports, i.e., the column indices of the m_j non-zero elements of $\Theta_{j,f}^c$ are denoted by $\mathcal{I}_{j,f}^c \subset \{0, \ldots, (N_j^X + 1)N_{full}\}$. The non-zero elements of the vector $\Theta_{j,f}^c$ are denoted by $\boldsymbol{\theta}_{j,f}^c$, and the sparse prediction model can be written as,

$$\hat{\mathbf{y}}_{j,f}^c = \mathbf{D}_{j,f}^c \boldsymbol{\Theta}_{j,f}^c = \mathbf{D}_{j,f,\mathcal{I}_{j,f}^c}^c \boldsymbol{\theta}_{j,f}^c = \tilde{\mathbf{D}}_{j,f}^c \boldsymbol{\theta}_{j,f}^c,$$

where $\tilde{\mathbf{D}}_{j,f}^c$ is the submatrix of $\mathbf{D}_{j,f}^c$, containing only the columns with indices in $\mathcal{I}_{j,f}^c$. The design of the sets $\mathcal{I}_{j,f}^c$ can be handled by a variety of sparse recovery algorithms. In this work we have used the optimized orthogonal matching pursuit approach, and the implementation known as fast orthogonal least squares [41]. The sets $\mathcal{I}_{j,f}^c$ are transmitted using $\lceil((N_j^X + 1)N_{full} + 1)/8\rceil$ bytes for each set and the coefficients $\boldsymbol{\theta}_{j,f}^c$ using quantization as in equation (10). The sparse filter from WaSP [9] can be obtained by setting $N_r = 1$ and by using using only the first N_{full} columns of $\mathbf{D}_{j,f}^c$ together with the bias column of ones.

V. AUXILIARY CODECS

In this section we explain how auxiliary codecs are used to encode texture reference and residual data, and normalized disparity maps. We enumerate the steps for obtaining the prediction residuals, and describe how the prediction residuals are encoded using intra coding tools in WaSP and using inter-view coding tools in the proposed codec WaSPR. Then we describe the quantization procedure applied to the normalized disparity maps prior to their intra coding using JPEG 2000.

A. CODING OF THE REFERENCE TEXTURE DATA AND PREDICTION RESIDUALS

The texture data is encoded using HEVC and refers both to the reference views $i \in \{0, \ldots, N_{ref} - 1\}$ which correspond to the original sub-aperture images at $\mathbf{X}(t_i^R, s_i^R)$, as well as to prediction residual views $\mathbf{E}_j$ at view indices (t_j^I, s_j^I). Prediction residual view for intermediate view j is defined as,

$$E_j(v, u, c) = X(t_j^I, s_j^I, v, u, c) - X_{\text{pred}}^{\text{dec}}(t_j^I, s_j^I, v, u, c),$$

where $\mathbf{X}_{\text{pred}}^{\text{dec}}(t_j^I, s_j^I)$ is obtained after inpaiting and region-based sparse prediction. The prediction residual view $\mathbf{E}_j$ contains the remaining difference between the predicted and the original view after the completion of the prediction stage. The sign of the residual needs to be preserved during the encoding-decoding cycle and a level-shift operation is applied to $\mathbf{E}_j$, followed by a rounding step,

$$\hat{\mathbf{E}}_j = \lfloor(\mathbf{E}_j + (2^b - 1))/2\rfloor.$$

$\hat{\mathbf{E}}_j$ is then encoded with HEVC using rate $\mathcal{R}_j^{res}$. At the decoder the decoded residual is obtained as,

$$\mathbf{E}_j^{\text{dec}} = 2\hat{\mathbf{E}}_j - (2^b - 1).$$

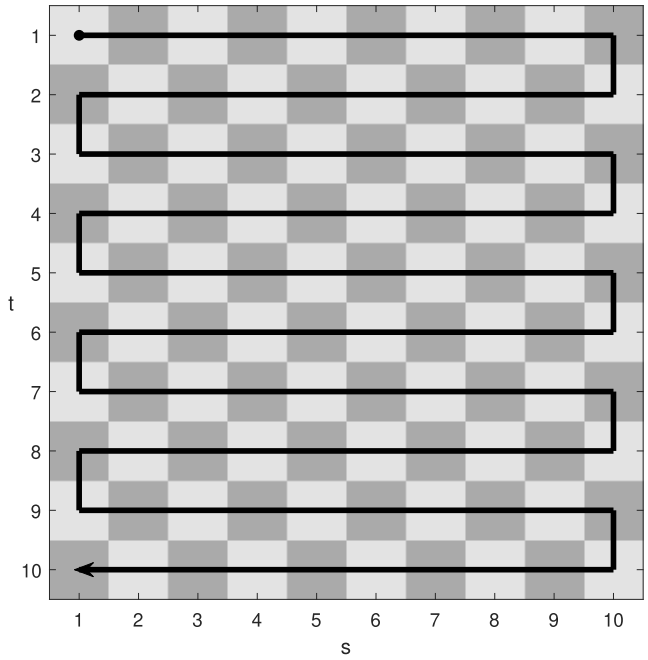

FIGURE 8. The scan-order for the inter-view coding of texture and residual views, indexed by (t, s), at each hierarchical level $H(t, s) \in \{1, \ldots, h_m\}$. Here the scan order is illustrated for a light field with $T = S = 10$. The same scan order is used at every hierarchical level, and view locations (t, s) which are not present in the current hierarchical level are skipped.

Quantizing the level-shifted residual by two ensures that the bit depth of $\hat{\mathbf{E}}_j$ is also b which is necessary when using an auxiliary codec with strict bit depth requirements. However, one bit of precision is lost.

At each hierarchical level the corresponding prediction residual views are obtained as a group and then encoded using standardized coding tools. In WaSP [9], [14] the coding of texture and residual views is performed using intra coding with JPEG 2000 [40]. Each texture reference view and prediction residual thus will have its own JPEG 2000 codestream. This approach fails to exploit the possible inter-view redundancies found in both texture and prediction residual views. For light fields obtained with an actual plenoptic camera, such as the Lytro Illum, some inter-view redundancy exists even after disparity based inter-view prediction. In the proposed codec WaSPR, the inter-view redundancy of texture and residual views is therefore further exploited using HEVC [20] with serpentine scan-order (i.e., Figure 8). The same scan-order is applied to both the texture residual views and the texture data of the initial set of reference views on the first hierarchical level.

B. CODING OF NORMALIZED DISPARITY

Normalized disparity $\tilde{\mathbf{Z}}$ is represented as a real valued quantity. For encoding of normalized disparity the WaSP(R) framework uses as an auxiliary codec JPEG 2000 [40]. While JPEG 2000 supports signed 32-bit floating point encoding, we choose to apply a simple quantization procedure to the normalized disparity $\tilde{\mathbf{Z}}$ prior to encoding. The quantization process first quantizes the floating point values of $\tilde{\mathbf{Z}}$ to integer range by applying a multiplication and truncation (rounding) operation and subsequently level shifts the quantized values into the positive range. For a given reference view $i \in \{0, \ldots, N_{ref} - 1\}$, the quantization operation for the normalized disparity data is,

$$\tilde{\mathbf{Z}}_{\mathrm{q}}(t_i^R, s_i^R, v, u) = \lfloor \tilde{\mathbf{Z}}(t_i^R, s_i^R, v, u) 2^{Q_{\tilde{Z}}} \rceil,$$

where $Q_{\tilde{Z}}$ is a quantization factor for the normalized disparity data and for all our experiments has the value 14. To ensure positive values, the level shifted and quantized normalized disparity is obtained using,

$$\tilde{\mathbf{Z}}_{\mathrm{ql}}(t_i^R, s_i^R, v, u) = \tilde{\mathbf{Z}}_{\mathrm{q}}(t_i^R, s_i^R, v, u) + M_{\tilde{Z}},$$

where $M_{\tilde{Z}}$ is defined as the absolute value of the smallest value found in the quantized normalized disparity,

$$M_{\tilde{Z}} = |\min_{\tilde{z} \in \tilde{\mathbf{Z}}_{\mathrm{q}}} \tilde{z}|.$$

After quantization and level-shifting, the normalized disparity $\tilde{\mathbf{Z}}_{\mathrm{ql}}$ is encoded with the selected auxiliary codec using the rate $\mathcal{R}_i^{\tilde{Z}}$. The decoded normalized disparity is obtained as,

$$\tilde{\mathbf{Z}}^{\mathrm{dec}} = (\tilde{\mathbf{Z}}_{\mathrm{ql}} - M_{\tilde{Z}}) / 2^{Q_{\tilde{Z}}}.$$

VI. HIERARCHICAL CODING OF VIEWS

One of the main difficulties of light field coding is the existence of occluded pixels between neighboring views. The disparity based warping scheme can be used to reduce the number of occluded pixels if the reference views can be selected to surround the target view both horizontally and vertically. A simple setup is to use as reference views the corners of the light field but in large disparity scenes, such as the HDCA, this approach produces inefficient prediction and is unable to solve the occlusion problem to the fullest.

In the hierarchical coding scheme of WaSP(R) the views are divided into disjoint subsets, each subset representing a hierarchical level. The encoder works in an hierarchical order, where the views on hierarchical level h are all encoded before encoding any of the views on the hierarchical level $h+1$. The views on the lower hierarchical levels operate as possible reference views to the views on the higher hierarchical levels, see Figure 9. As the encoder proceeds towards the higher hierarchical levels, the density of the reference views increases and the intermediate view prediction becomes more efficient. The decrease of the energy of the prediction residual leads to more efficient coding of the higher hierarchical levels. Therefore at the higher hierarchical levels the required image quality can be achieved with smaller codelengths for residuals. The hierarchical configurations for three light field images greek, I01_Bikes and Set2 are shown in Figure 10. The dense light fields greek and I01_Bikes use six hierarchical levels with just one view residing at the lowest hierarchical level. Full reconstruction of subsequent hierarchical levels is still possible due to the density of the light field and substantially difficult occlusions do not occur. For the sparse light field

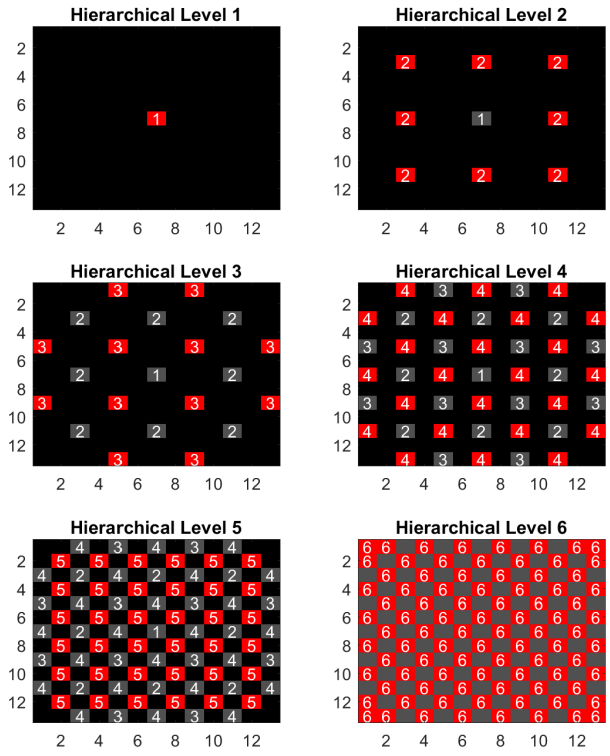

FIGURE 9. The split of the views obtained from a plenoptic camera into six hierarchical levels. At each hierarchical level the views marked in red are to be coded conditional on the reference views (marked in gray) using the proposed prediction scheme. At the lowest hierarchical level 1 no reference views are available and thus the center view is encoded without prediction using the auxiliary codec only. Already at hierarchical level 2 a reference view is available, and the intermediate views (in red) can be encoded conditional on the reference views. The number and density of reference views grows as a function of the hierarchical level increasing the efficiency of the 4D prediction.

Set2 the lowest hierarchical level occupies the corners and the center of the camera array. The combined prediction of the warped five reference views is enough to cover the possibly occluded areas resulting from the warping of the individual reference views.

The hierarchical scheme is not used for the synthesis of the normalized disparity. Since no prediction residuals are obtained for the synthesized disparity views, the hierarchical propagation of prediction errors may lead to dramatic decrease in the accuracy of the synthesized disparity views. For this reason, the normalized disparity views are always synthesized from the lowest hierarchical level.

A. AUXILIARY CODEC RATE ALLOCATION BETWEEN HIERARCHICAL LEVELS

With the exception of the very low bit rates (< 0.01 bpp) most of the code length in the proposed codec is used for the codestreams of the auxiliary codecs. Because of this a rate allocation scheme is needed to efficiently distribute the bit budget between different hierarchical levels. In Algorithm 2 we present an iterative algorithm which divides the joint bit allocation of h_m hierarchical levels into $h_m - 1$ number of optimizations.

The main procedure $\boldsymbol{\mathcal{B}} = \text{BitAllocation}(B)$ takes as an input parameter B the total number of bits targeted by the encoder for coding of the texture component given by (5) and (6). The output $\boldsymbol{\mathcal{B}}$ is a vector containing the bit budget for the auxiliary codec codestreams of each hierarchical level $h \in \{1, \ldots, h_m\}$. The algorithm begins by allocating the full bit budget B for the lowest level, and assigning 0 bits for the rest of the hierarchical levels. If $h_m = 1$ the procedure ends and the full bit budget is used at the lowest hierarchical level by setting $\mathcal{B}(1) = B$. If $h_m > 1$ the algorithm proceeds in iterative fashion to obtain the values $\mathcal{B}(h)$ for $\forall h \in \{1, \ldots, h_m\}$ by maximizing the fidelity of the decoded light field.

The function $\mathcal{D}(\boldsymbol{\mathcal{B}}, h)$ encodes the light field using the rate allocation $\boldsymbol{\mathcal{B}}$ and obtains the fidelity of the decoded light field for the levels up to and including h. The fidelity criterion can be for example the peak signal-to-noise ratio (PSNR) or the structural similarity index [42].

The first of the $h_m - 1$ iterations obtains the optimal splitting of the B bits between levels $h = 2$ and $h - 1 = 1$ by maximizing the fidelity $D = \mathcal{D}(\boldsymbol{\mathcal{B}}, h)$ on lines 4-9 in which the fidelity is evaluated for every split parameter γ in the set Γ_B. The split parameter γ is used in the function $\boldsymbol{\mathcal{B}} = \text{ReAllocate}(\boldsymbol{\mathcal{B}}_{h-1}, h, \gamma)$ which scales the bit allocations for levels $i \leq h - 1$ by γ, and assigns $(1 - \gamma)B$ bits to the level h. In the case of $h = 2$ the function simply applies a γ controlled division or split of the B bits between the two successive hierarchical levels. At line 8, after obtaining the fidelity of the decoded light field for a set of bit allocations for $\gamma \in \Gamma_B$, the algorithm picks the split parameter γ_o which yields the highest fidelity of the decoded light field. The B bits are accordingly partitioned into the vector $\boldsymbol{\mathcal{B}}_h = \boldsymbol{\mathcal{B}}_2$ using the reallocation function $\boldsymbol{\mathcal{B}}_2 = \text{ReAllocate}(\boldsymbol{\mathcal{B}}_1, 2, \gamma_o)$. In the next iteration ($h = 3$), the bits of levels $h - 2 = 1$ and $h - 1 = 2$ are again scaled using γ_o and the remaining $(1 - \gamma_o)B$ bits are used for the level $h = 3$, maximizing the fidelity of the decoded light field for the levels $h \in \{1, 2, 3\}$, producing the vector $\boldsymbol{\mathcal{B}}_3$. The iteration ends once $\boldsymbol{\mathcal{B}}_{h_m}$ is obtained.

For larger light fields, such as Set2, applying the hierarchical rate allocation scheme for all views becomes a rather slow process due to each hierarchical level being successively encoded several times. By assuming that the view prediction performance inside a hierarchical level is rather uniform, a speed-up can be obtained by applying the rate allocation for only a subset of the views. This strategy was used in obtaining the rate allocation for Set2 with white crosses in Figure 10 illustrating the subset of view used.

VII. CODESTREAM

The codestream is divided into two parts: the header section, and the codestreams for each view. The header section contains necessary parameters for decoding to take place such as the number of views in the codestream, the sub-aperture image height V and width U, the offset parameter $M_{\tilde{Z}}$ used in the quantization of the normalized disparity, and information about the colorspace used. Following the header section is a series of sub-aperture views, each having their own set

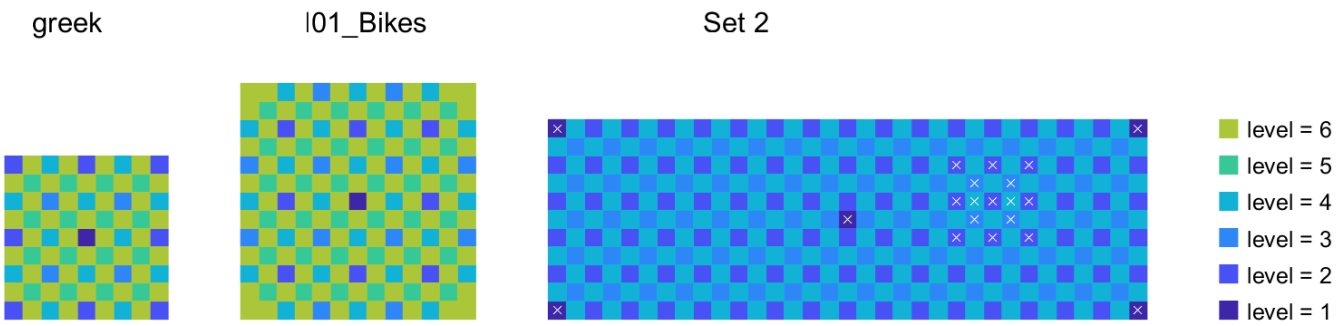

FIGURE 10. Hierarchy matrices H for images greek, I01_Bikes and Set2. For the dense light fields a single texture and normalized disparity reference at the center of the camera array in lowest hierarchical level is enough for good quality view prediction in the subsequent hierarchical levels. For sparse light fields the lowest hierarchical level occupies the extremities and the center of the camera array providing occlusion free prediction over practically the whole light field. For Set2 the white crosses indicate the subset of views which were used in rate allocation.

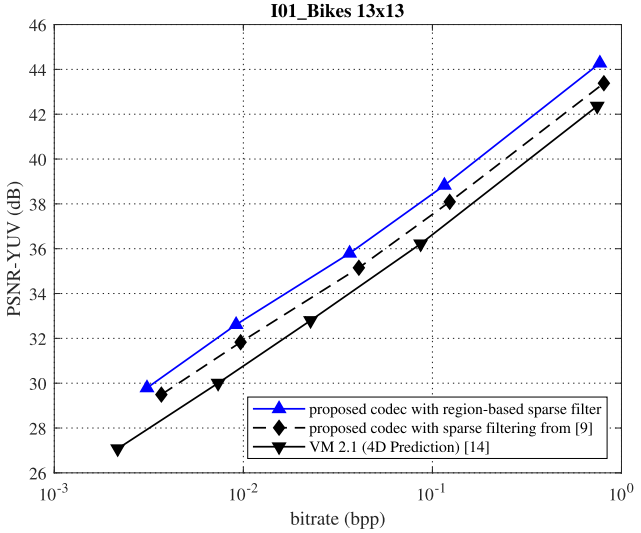

FIGURE 11. Comparing the rate-distortion performance of the proposed scheme against the 4D prediction mode (i.e., WaSP) of VM 2.1 [14]. The dashed line represents the proposed scheme with the sparse filtering scheme of the VM 2.1 (WaSP). This reveals, that the proposed inter-coding scheme for residuals improves over the JPEG 2000 based intra coding used in VM 2.1 (WaSP). The solid blue line represents the proposed scheme using the novel region-based sparse filter and the new inter-coding scheme for residuals. Substantial improvement can be observed against the sparse filtering scheme of VM 2.1 (WaSP, the dashed line).

Algorithm 2 Bit Allocation for Hierarchical Levels $h = 1, \ldots, h_m$

procedure $\mathcal{B} = \text{BitAllocation}(B)$
  $\mathcal{B}_1(1) = B$
  $\mathcal{B}_1(h) = 0, \ \forall h \in \{2, \ldots, h_m\}$
  for $h = 2, \ldots, h_m$ **do**
    for $\gamma \in \Gamma_B$ **do**
      $\mathcal{B} = \text{ReAllocate}(\mathcal{B}_{h-1}, h, \gamma)$
      $D_\gamma = \mathcal{D}(\mathcal{B}, h)$
    end for
    $\gamma_o = \text{argmax}_\gamma \, D_\gamma$
    $\mathcal{B}_h = \text{ReAllocate}(\mathcal{B}_{h-1}, h, \gamma_o)$
  end for
  return $\mathcal{B} = \mathcal{B}_{h_m}$
end procedure
function $\mathcal{B} = \text{ReAllocate}(\mathcal{B},h,\gamma)$
  $B = \sum_{i=1}^{h-1} \mathcal{B}(i)$
  for $k = 1, \ldots, h-1$ **do**
    $\mathcal{B}(k) = \gamma \mathcal{B}(k)$
  end for
  $\mathcal{B}(h) = (1-\gamma)B$
  return $\mathcal{B}$
end function

of camera parameters, prediction parameters, and both texture and normalized disparity codestreams from the auxiliary codecs.

The parameters in the codestreams for reference and intermediate views are different, since the reference views lack the view prediction parameters and only contain the sub-aperture view attributes, such as camera parameters, view indices, and the codestreams from the auxiliary codec. The number of parameters used by the view merging algorithm increases rapidly as a function of the number of reference views but we have used rarely more than four references. The view merging parameters are followed by the parameters of the region-based sparse filter, which include the number of regions used, the filter size, the filter order, the location of the regressors in the template signaled as a bit mask, and finally the filter coefficients themselves. In WaSPR the view parameters are further encoded with Deflate [43]. The codestream of the prediction residual is appended directly as the bytes given by the auxiliary codec.

VIII. EXPERIMENTAL RESULTS

In this section the performance evaluation of the proposed codec is discussed.

A. DATASETS

The R-D results of the proposed encoder are reported on datasets used by the JPEG Pleno standardization activity [19]. The datasets tested are the ones provided by Multimedia Signal Processing Group of Ecole Polytechnique Fédérale de Lausanne (EPFL) [44], Heidelberg Collaboratory for Image Processing (HCI) [45], and Fraunhofer Institute for Integrated Circuits [46]. The datasets as used in this paper are available

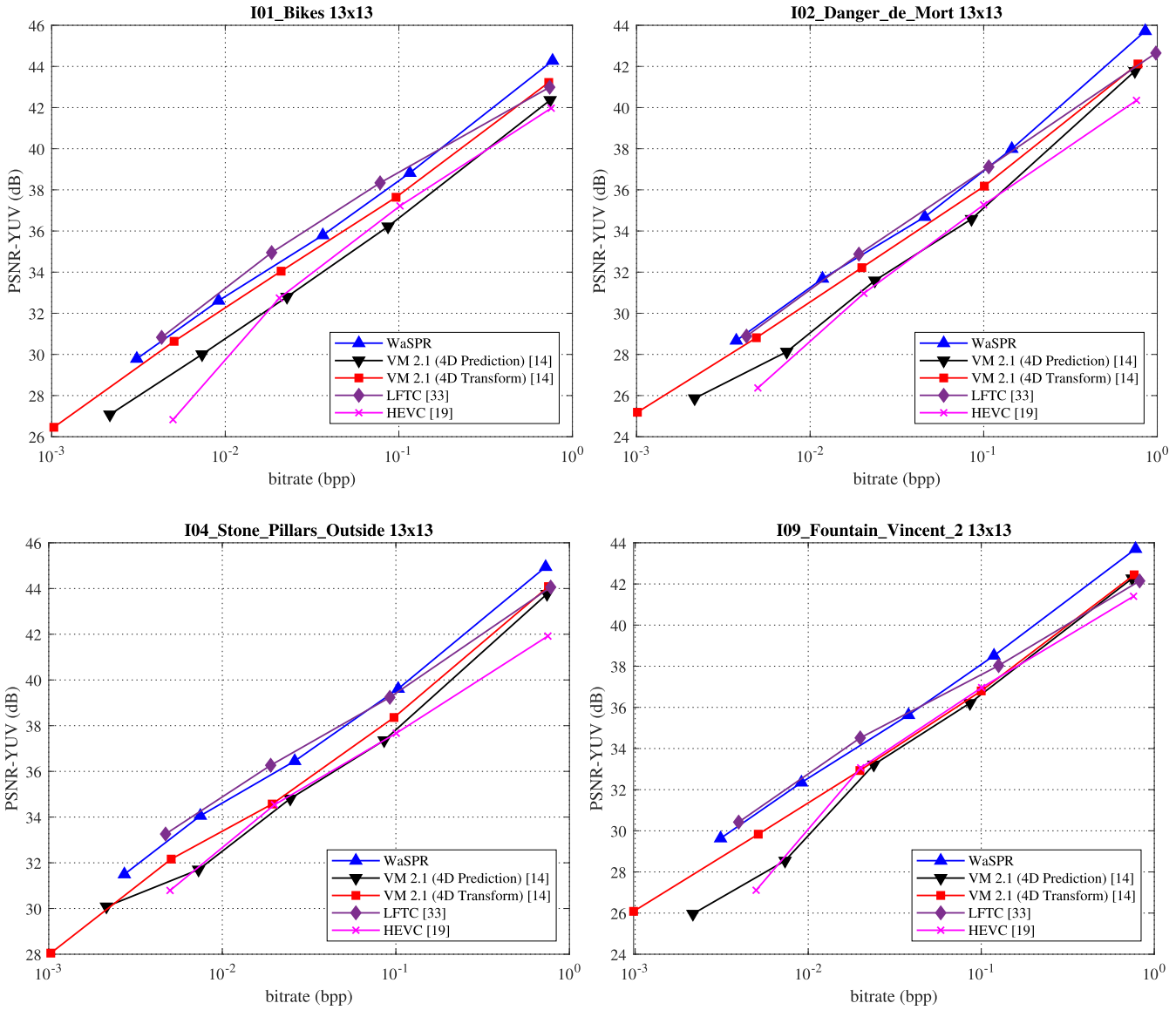

FIGURE 12. Rate-distortion performance of the proposed codec on plenoptic images. Comparisons are made against VM 2.1 (WaSP and MuLE) [14], LFTC [33], and HEVC anchors [19], under the conditions specified in the CTC [19].

in JPEG Pleno database [47]. Following the conventions of the JPEG Pleno light field standardization the sample bit depth for the texture views is set to 10 bits and the 8 bit datasets have been scaled to 10 bits per sample. According to the CTC we use a sparse set of 11×33 views from Set2 [46] which are further cropped to the central 1080×1920 rectangle of pixels. However, we also demonstrate the performance of the proposed codec on the full Set2 in a comparison against [16].

The EPFL dataset provides images taken with Lytro Illum plenoptic camera which have been post-processed into sub-aperture images using [18]. The Lytro Illum images have a small disparity range (less than 15 pixels between extreme views) similar to those in the synthetic dataset provided by HCI, where 3D rendering software was used to generate views similar to those obtained using a plenoptic camera. The HDCA dataset by IIS has several hundreds of pixels of inter-view disparity, making it a very different dataset, which is not tackled in the existing literature, excepting [16].

B. QUALITY METRICS

We demonstrate the performance of our codec according to the quality metrics used by the JPEG Pleno standardization using the Matlab implementations in [19]. The quality metrics are all evaluated in the YCbCr color space including a weighted PSNR YCbCr and PSNR Y. The weighted PSNR YCbCr is computed as $(6 \cdot \mathrm{PSNR(Y)} + \mathrm{PSNR(Cb)} + \mathrm{PSNR(Cr)})/8$.

C. CONFIGURATION OF THE AUXILIARY CODECS

The texture reference, residual, and normalized disparity coding can be performed with a variety of auxiliary codecs.

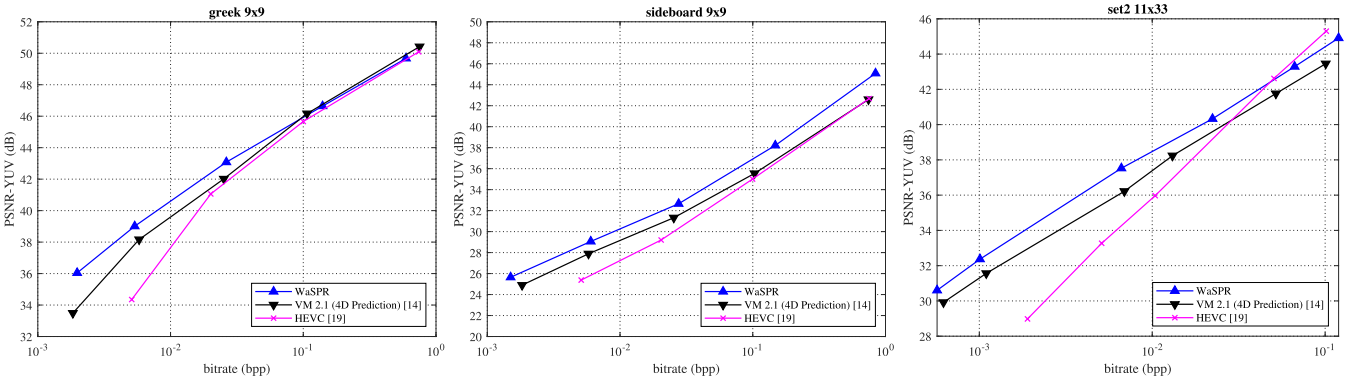

FIGURE 13. Rate-distortion performance of the proposed codec on the densely sampled light fields Greek and Sideboard, and on the sparsely sampled light field Set2. Comparisons are made against VM 2.1 (WaSP) [14] and HEVC anchors [19], under the conditions specified in the CTC [19].

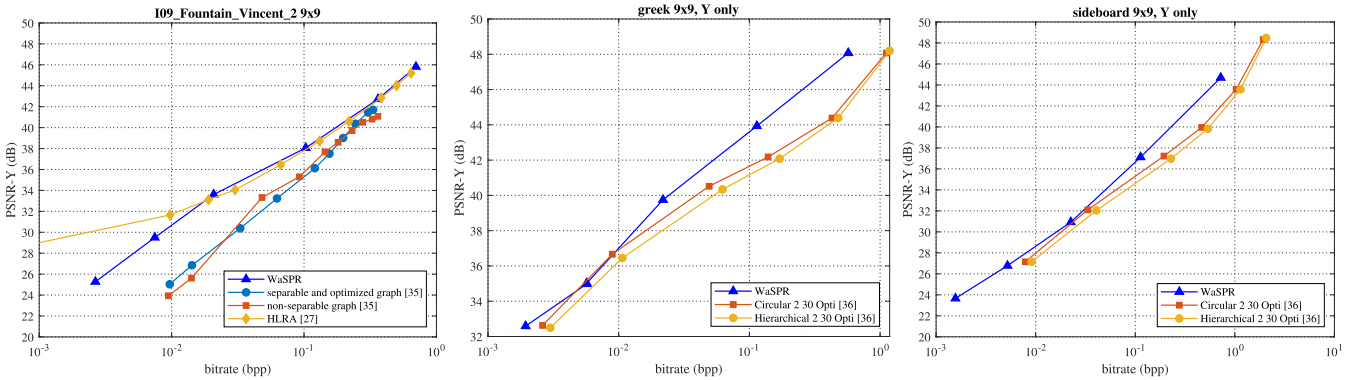

FIGURE 14. Rate-distortion performance of the proposed codec on the plenoptic image I09_Fountain_Vincent_2 and on the densely sampled light fields Greek and Sideboard. Comparisons are made against GBT [35], HLRA [27], and FDL [36], under the conditions specified in the publications [27], [35], [36].

Technically any image/video codec with suitable support for the given bit depth and resolution can be used. We have experimented with two auxiliary codecs: JPEG 2000 for normalized disparity and HEVC for texture and residual.

For HEVC coding we use HEVC Test Model version 16.20 [48] with a group of pictures structure of size 8 for EPFL and HCI datasets and intra coding for the sparsely sampled light field Set2. The full 444 chrominance sampling is used for the reference views and 400 sub-sampling for the residual views; we let our prediction to fully handle the chrominance prediction of the intermediate views. The QP parameter of each hierarchical level was set to closely match the rates obtained from the rate-allocation scheme of Section VI-A.

For JPEG 2000 we use the free implementation of Kakadu version 7.10.2 [49]. We used the command line parameter `-no_info` to remove a codestream marker segment which is not necessary for decoding, `-no_weights` to allow for the direct minimization of mean squared error instead of visual quality, `-precise` to force 32-bit processing during transforms increasing precision, and `Clevels=6` for setting the number of wavelet decomposition levels to six which was considered optimal.

D. RATE-DISTORTION PERFORMANCE EVALUATION

We compare the R-D performance of the proposed codec against JPEG Pleno VM 2.1 [14], the light field translation codec (for which we use the acronym LFTC) [33], HEVC in pseudo-temporal sequence [19], HLRA [27], GBT [35], FDL [36] and the HDCA compression method of [16]. In Figures 11-15 the R-D comparisons are provided following the common test conditions [19] with the exception that the evaluations against HLRA, GBT, and FDL were obtained using 9×9 views. Furthermore, when evaluating against FDL, only the Y component was encoded as was also done in [36]. The R-D results for the codecs were obtained from their corresponding publications and authors, with the exception of LFTC, for which we used the implementation provided in [50] and the parameters offered in [33].

From the Figure 11 it can be seen that for the densely sampled light fields the gain of WaSPR compared to WaSP comes both from the higher efficiency of HEVC compared to JPEG 2000, and from the region-based sparse filtering; the multiple filters being restricted to the individual disparity regions offer a far more coherent set of predictors while the filter in [9] attempted to combine everything in a single sparse predictor.[1] The region based sparse filter of WaSPR has an increased computational complexity over the sparse filter used in WaSP. Regarding the results reported in Figure 11, we observed an average increase in the encoder execution time of roughly 20% for high rates, and 84% for low rates,

[1]For visual comparison see http://www.cs.tut.fi/~astolap/WaSPR

when comparing the new region based sparse filter of WaSPR against the earlier sparse filter of WaSP. However, we note that our implementation of the proposed codec has not been thoroughly optimized with respect to execution time.

As shown in Figures 12-13, the proposed codec performs better than both of the VM 2.1 encoding modes on the JPEG Pleno datasets. For encoding the EPFL dataset the average Bjontegaard [51] rate reductions for PSNR YCbCr are -48.1% and -27.8% against the 4D prediction (WaSP) and 4D transform (MuLE) modes of VM 2.1 respectively. For the HCI dataset the proposed codec obtains -28.0% rate reduction against VM 2.1, and with the sparsely sampled light field Set2 the proposed codec obtains -53.4% rate reduction.

At low and medium rates the proposed codec offers quite similar performance when compared against LFTC with the proposed scheme obtaining substantial improvement at higher rates as shown in Figure 12. Both WaSPR and LFTC are somewhat similar in nature; HEVC is used for coding of reference views and a segmentation based linear predictor is used to predict the intermediate views. Both methods rely on a segmentation obtained from disparity based labeling; LFTC's segmentation is build on quad-tree partitioning using depth boundaries while WaSPR uses more arbitrary shaped regions discriminated by their disparity values. One likely explanation for the R-D performance differences between WaSPR and LFTC is the choice of segmentation, and the difference in residual coding with the latter using a low-rank approximation strategy. A careful study combining best features from both approaches might obtain superior R-D performance on all rate points.

Compared to GBT the proposed scheme obtains better R-D results with the gain being largest at the lowest rates, as shown in Figure 14. Similarly the proposed codec improves over FDL with the R-D improvements becoming largest at the higher rates. HLRA obtains better performance at the lowest rates but at higher rates the R-D performance becomes almost identical to the proposed method.

In Figure 15 the 21×97 views of Set2 were encoded at the full 2160×3840 resolution. Similar to [16] only the subset of 11×33 views use residual coding and the rest of the 1674 views were predicted only using the hierarchical sparse prediction scheme. The rate-allocation scheme was applied on the cropped 11×33 views and the obtained QP parameters were manually adjusted for a better performance. Both the proposed method and [16] use warping of the texture using normalized disparity maps and residual coding using existing well-established image coding solutions. The method of [16] places more emphasis on the modeling of the normalized disparity field, with a base-anchored mesh model used to infer disoccluded areas via backfilling strategy. We infer the full set of normalized disparity maps with direct warping and merging of the five reference normalized disparity maps with region based correction using sparse filtering during texture prediction. The method of [16] also uses BPA-DWT variant of EBCOT [52] for coding of normalized disparity, which has been shown to improve WaSP performance on HDCA images

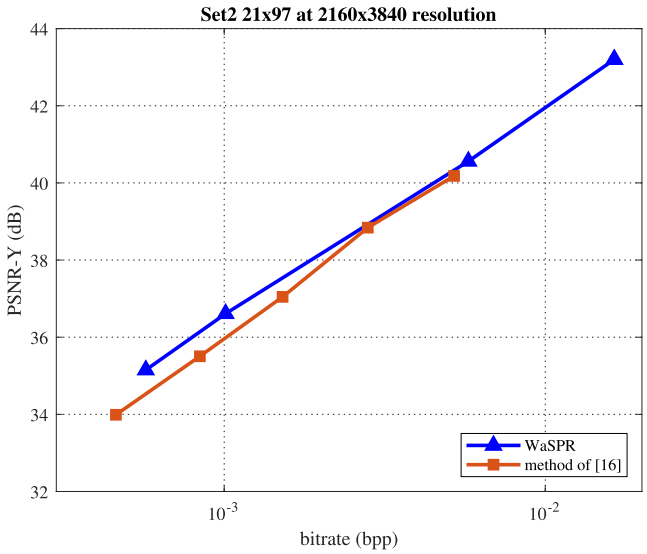

FIGURE 15. Rate-distortion performance for Set2 using 21×97 views at 2160×3840 resolution. The performance of the proposed method is compared to [16].

[37], while our method still relies on baseline JPEG 2000 for coding of normalized disparity.

The rate allocation between texture/residual, normalized disparity, and prediction parameter data varies depending on the size of the light field and on the rate targeted by the encoder, with the texture/residual using from 43% to 97% of the total rate. At the highest rates most of the bits are used for texture reference and residual coding, while at the lowest rates the prediction parameters use an equal or larger fraction of the encoded bits. The normalized disparity data uses between $0.04 - 32.50\%$ of the total rate.

IX. CONCLUSION AND DISCUSSION

In this paper we have described a new light field coding scheme WaSPR based on the WaSP framework with rate-distortion performance shown to exceed both WaSP and MuLE modes of the JPEG Pleno VM 2.1. The proposed codec performs well against recent state-of-the-art light field codecs for both densely and sparsely sampled light fields. We have provided a detailed description and discussion of the common parts of WaSP and WaSPR, provided a rate allocation algorithm for hierarchical coding of light fields, introduced inter-view coding of texture reference and residual views into the WaSP framework, and proposed a new region-based sparse prediction scheme shown to significantly reduce the magnitude of the prediction error resulting from direct disparity based view prediction. The WaSPR software [53] has been made publicly available for the research community.

REFERENCES

[1] G. Lippmann, "Épreuves réversibles donnant la sensation du relief," *J. Phys. Theor. Appl.*, vol. 7, no. 1, pp. 821–825, 1908.
[2] A. Gershun, "The light field," *J. Math. Phys.*, vol. 18, nos. 1–4, pp. 51–151, 1939.
[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn. SIG-*

GRAPH. New York, NY, USA, 1996, pp. 43–54.
[4] M. Levoy, "Light fields and computational imaging," *Computer*, vol. 39, no. 8, pp. 46–55, Aug. 2006.
[5] R. Ng, M. Levoy, and M. Bredif, G. Duval, M. Horowitz, P. Hanrahan, "Light field photography with a hand-held plenoptic camera," Stanford Univ. Comput. Sci., Stanford, CA, USA, Tech. Rep. CSTR 2005-02, Apr. 2005.
[6] T.-J. Li, S. Li, Y. Yuan, Y.-D. Liu, C.-L. Xu, Y. Shuai, and H.-P. Tan, "Multi-focused microlens array optimization and light field imaging study based on Monte Carlo method," *Opt. Express*, vol. 25, no. 7, p. 8274, 2017.
[7] I. Tabus, P. Helin, and P. Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4567–4571.
[8] P. Astola and I. Tabus, "Light field compression of HDCA images combining linear prediction and JPEG 2000," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 1860–1864.
[9] P. Astola and I. Tabus, "WaSP: Hierarchical warping, merging, and sparse prediction for light field image compression," in *Proc. 7th Eur. Workshop Vis. Inf. Process. (EUVIP)*, Nov. 2018, pp. 1–6.
[10] *JPEG Pleno Light Field Coding VM 1.0*, document N80028, ISO/IEC JTC 1/SC29/WG1 JPEG, Jul. 2018.
[11] *JPEG Pleno Light Field Coding VM 1.1*, document N81052, ISO/IEC JTC 1/SC29/WG1 JPEG, Oct. 2018.
[12] *Verification Model Software Version 2.1 on JPEG Pleno Light Field Coding*, document N83034, Mar. 2019.
[13] P. Schelkens, P. Astola, E. A. B. da Silva, C. Pagliari, C. Perra, I. Tabus, and O. Watanabe, "JPEG Pleno light field coding technologies," in *Applications of Digital Image Processing XLII*, vol. 11137, A. G. Tescher and T. Ebrahimi, Eds. Bellingham, WA, USA: SPIE, 2019, pp. 391–401, doi: 10.1117/12.2532049.
[14] C. Perra, P. Astola, E. A. B. da Silva, H. Khanmohammad, C. Pagliari, P. Schelkens, and I. Tabus, "Performance analysis of JPEG Pleno light field coding," in *Applications of Digital Image Processing XLII*, vol. 11137, A. G. Tescher and T. Ebrahimi, Eds. Bellingham, WA, USA: SPIE, 2019, pp. 402–413, doi: 10.1117/12.2528391.
[15] P. Astola. *WaSP—Light Field Compression*. Accessed: Nov. 7, 2019. [Online]. Available: https://github.com/astolap/WaSP
[16] D. Rüfenacht, A. T. Naman, R. Mathew, and D. Taubman, "Base-anchored model for highly scalable and accessible compression of multi-view imagery," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3205–3218, Jul. 2019.
[17] M. Le Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1981–1993, Apr. 2018.
[18] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2013, pp. 1027–1034.
[19] *JPEG Pleno Light Field Common Test Conditions 3.3*, document N84049, ISO/IEC JTC 1/SC29/WG1 JPEG, Jul. 2019.
[20] 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.
[21] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of Lytro-Illum light fields," in *Proc. Int. Conf. Image Process. Theory, Tools Appl.*, Nov. 2015, pp. 494–497.
[22] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.
[23] W. Ahmad, R. Olsson, and M. Sjöström, "Interpreting plenoptic images as multi-view sequences for improved compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4557–4561.
[24] H. Amirpour, M. Pereira, and A. Pinheiro, "High efficient snake order pseudo-sequence based light field image compression," in *Proc. Data Compress. Conf.*, Sep. 2017, p. 397.
[25] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.
[26] M. B. de Carvalho, M. P. Pereira, G. Alves, E. A. B. da Silva, C. L. Pagliari, F. Pereira, and V. Testoni, "A 4D DCT-based lenslet light field codec," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 435–439.
[27] X. Jiang, M. Le Pendu, R. A. Farrugia, and C. Guillemot, "Light field compression with homography-based low-rank approximation," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1132–1145, Oct. 2017.
[28] W. Ahmad, S. Vagharshakyan, M. Sjöström, A. Gotchev, R. Bregovic, and R. Olsson, "Shearlet transform based prediction scheme for light field compression," in *Proc. Data Compress. Conf.*, Mar. 2018, p. 396.
[29] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, Jan. 2016.
[30] T.-H. Tran, Y. Baroud, Z. Wang, S. Simon, and D. Taubman, "Light-field image compression based on variational disparity estimation and motion-compensated wavelet decomposition," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3260–3264.
[31] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.
[32] I. Viola, H. P. Maretic, P. Frossard, and T. Ebrahimi, "A graph learning approach for light field image compression," in *Applications of Digital Image Processing XLI*, vol. 10752, A. G. Tescher, Ed. Bellingham, WA, USA: SPIE, 2018, pp. 126–137, doi: 10.1117/12.2322827.
[33] B. Hériard-Dubreuil, I. Viola, and T. Ebrahimi, "Light field compression using translation-assisted view estimation," in *Proc. Picture Coding Symp. (PCS)*, 2019, p. 5.
[34] E. Dib, M. Le Pendu, X. Jiang, and C. Guillemot, "Super-ray based low rank approximation for light field compression," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, pp. 369–378.
[35] M. Rizkallah, X. Su, T. Maugey, and C. Guillemot, "Geometry-aware graph transforms for light field compact representation," *IEEE Trans. Image Process.*, vol. 29, pp. 602–616, 2020, doi: 10.1109/TIP.2019.2928873.
[36] E. Dib, M. L. Pendu, and C. Guillemot, "Light field compression using Fourier disparity layers," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3751–3755.
[37] R. Mathew and D. Taubman, "WaSP encoder with breakpoint adaptive DWT coding of disparity maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 3172–3176.
[38] L. A. Thomaz, J. M. Santos, P. Astola, S. M. M. de Faria, P. A. A. Assuncao, M. B. de Carvalho, E. A. B. da Silva, C. L. Pagliari, I. Tabus, M. P. Pereira, G. O. Alves, F. Pereira, V. Testoni, P. G. Freitas, and I. Seidel, "Visually lossless compression of light fields," in *Proc. Eur. Light Field Imag. Workshop*, Jun. 2019.
[39] J. M. Santos, P. A. A. Assuncao, L. A. D. S. Cruz, L. M. N. Tavora, R. Fonseca-Pinto, and S. M. M. Faria, "Lossless compression of light fields using multi-reference minimum rate predictors," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, pp. 408–417.
[40] D. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Boston, MA, USA: Kluwer, 2002.
[41] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Process.*, vol. 43, no. 7, pp. 1713–1715, Jul. 1995.
[42] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[43] *Gzip*. Accessed: Nov. 7, 2019. [Online]. Available: https://www.gnu.org/software/gzip/
[44] T. E. M. Rerabek, "New light field image Dataset," in *Proc. 8th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, 2016.
[45] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A Dataset and Evaluation Methodology for Depth Estimation on 4D Light Fields," in *Comput. Vision—ACCV*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Berlin, Germany: Springer, 2017, pp. 19–34.
[46] M. Ziegler, R. O. H. Veld, J. Keinert, and F. Zilly, "Acquisition system for dense lightfield of large scenes," in *Proc. 3DTV Conf., True Vis. Capture, Transmiss. Display 3D Video (3DTV-CON)*, Jun. 2017, pp. 1–4.
[47] ISO/IEC JTC 1/SC29/WG1 JPEG. (2019). *JPEG Pleno Database*. [Online]. Available: https://jpeg.org/plenodb/lf/pleno_lf/
[48] *High Efficiency Video Coding Test Model (HM)*. Accessed: Nov. 7, 2019. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[49] *Kakadu Software*. Accessed: Nov. 7, 2019. [Online]. Available: http://www.kakadusoftware.com
[50] Multimedia Signal Processing Group at Swiss Federal Institute of Technology (EPFL). *Light-Field-Translation-Codec*. Accessed: Nov. 7, 2019. [Online]. Available: https://github.com/mmspg/light-field-translation-codec
[51] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD Curves*, document VCEG-M33, ITU-T VCEG Meeting, 2001.
[52] R. Mathew, D. Taubman, and P. Zanuttigh, "Scalable coding of depth maps with R-D optimized embedding," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1982–1995, May 2013.
[53] P. Astola. *WaSPR—Light Field Compression*. Accessed: Nov. 7, 2019. [Online]. Available: https://github.com/astolap/WaSPR

PEKKA ASTOLA received the M.Sc. degree in signal processing and communications engineering from the Tampere University of Technology, Tampere, Finland, in 2013. He is currently pursuing the Ph.D. degree with the Computing Sciences Unit, Tampere University, under the supervision of Prof. I. Tabus. Since 2017, he has been actively participating in the working group ISO/IEC JTC1/SC29/WG1 developing the JPEG Pleno light field compression standard. He is the author of the JPEG Pleno Light Field Verification Model Software versions 1.0 and 1.1. His research interest includes light field image compression and processing. He was a corecipient of the 2016 3DTV Best Paper Award, the 2016 ISMVL Outstanding Paper Award, and the ICIP 2017 Light Field Image Coding Challenge Award.

IOAN TABUS received the Ph.D. degree (Hons.) from the Tampere University of Technology, Finland, in 1995. He held teaching positions at the Department of Control and Computers, Politehnica University of Bucharest, from 1984 to 1995. Since 1996, he has been a Senior Researcher and a Professor with the Department of Signal Processing, Tampere University of Technology, since January 2000, which was merged into the Tampere University, in 2019. He is the coauthor of two books and more than 250 publications in the fields of signal compression, image processing, bioinformatics, and system identification. His research interests include light field image processing, plenoptic image compression, point clouds compression, audio, image, and data compression, genomic signal processing, and statistical signal processing. He was a Corecipient of 1991 Train Vuia Award of Romania, the 2001 NSIP Best Paper Award, the 2004 NORSIG Best Paper Award, the 2016 3DTV Best Paper Award, and the ICIP 2017 Light Field Image Coding Challenge Award. He served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and *Signal Processing* (EURASIP). He has served as a Guest Editor of special issues for the *IEEE Signal Processing Magazine*, *Signal Processing* (EURASIP), and the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. He was the Editor-in-Chief of the *EURASIP Journal on Bioinformatics and Systems Biology*, from 2006 to 2014.