Lossless Coding of Light Fields based on 4D Minimum Rate Predictors

João M. Santos, Student Member, IEEE, Lucas A. Thomaz, Member, IEEE,
Pedro A. A. Assunção, Senior Member, IEEE, Luís A. da Silva Cruz, Senior Member, IEEE, Luís Távora, and Sérgio M. M. Faria, Senior Member, IEEE

Abstract—Common representations of light fields use four-dimensional data structures, where a given pixel is closely related not only to its spatial neighbours within the same view, but also to its angular neighbours, co-located in adjacent views. Such structure presents increased redundancy between pixels, when compared with regular single-view images. Then, these redundancies are exploited to obtain compressed representations of the light field, using prediction algorithms specifically tailored to estimate pixel values based on both spatial and angular references. This paper proposes new encoding schemes which take advantage of the four-dimensional light field data structures to improve the coding performance of Minimum Rate Predictors. The proposed methods expand previous research on lossless coding beyond the current state-of-the-art. The experimental results, obtained using both traditional datasets and others more challenging, show bit-rate savings no smaller than 10%, when compared with existing methods for lossless light field compression.

Index Terms—Light field compression, 4D prediction, 4D partition, Lossless coding, Medical Imaging

I. INTRODUCTION

LIGHT field (LF) imaging, initially proposed by Lippmann in 1908 [1], promises to provide superior immersive multimedia experiences for users and content producers. Light fields fully characterise the flow of light in every direction and point in space. With appropriate sampling, this function can be translated to imaging data that can represent both traditional 2D spatial information and also the directionality of light rays, i.e., angular information, reaching the camera sensors. Applications of this technology range from the entertainment industry, such as virtual and augmented reality [2] and 3D movies and television [3], to more critical tasks as medical imaging [4]–[6], for instance. These various applications are only possible due to the characteristics of LFs, which represent increased visual information when compared to traditional 2D images. Such extra information enables a panoply of post-processing operations, like the extraction of depth maps [7]–[9], or image rendering with different viewing perspectives [10], or post-capture refocusing [11].

Developments in LF acquisition systems technology are expected to make LF imaging available to a broader range of users, both industrial and domestic. However, one of the potential deterrents of adoption of LF imaging is the huge amount of storage capacity and bandwidth required to handle all the data required by this technology. In consideration of this, both the Joint Photographic Experts Group (JPEG) [12] and the Moving Picture Experts Group (MPEG) [13] started standardisation initiatives with the objective of developing efficient encoding techniques of LFs, ranging from lossy to lossless compression.

Light field compression has been the subject of intensive research by the scientific community, but most of this effort has been directed towards lossy coding, as evidenced by the extensive survey of [14]. However, not all applications can afford the loss of information, in particular, applications such as medical imaging or precision measurements in industry need high fidelity in the reconstruction of images, requiring lossless and near-lossless performance. Lossless coding, on the other hand, is a less explored topic in LF compression. Despite the JPEG-Pleno call for proposals [12] asking for both lossy and lossless coding solutions, from the selected algorithms for the standard only WaSP [15] is currently capable of achieving lossless compression. Even so, in the JPEG-Pleno standardisation activity experiment [16] it was shown that a non-standard encoder [17], based on Minimum Rate Predictors (MRP) [18], achieves higher lossless compression efficiency. The underlying principles of MRP consist of estimating the amount of information conveyed by prediction errors resulting from predictors, which are specifically designed to minimise the rate rather than the prediction error itself, as usually done in lossy encoding schemes.

LFS are usually represented as 4D functions, with two spatial dimensions accounting for the 2D spatial information, also present in regular images (v, u), and two angular dimensions accounting for the different viewpoints that are also captured in LFs (t, s). 4D LFs present increasing redundancy, when compared with 2D images, that can be exploited to provide higher compression efficiency than that provided by encoders which cannot exploit all this extra information. This paper exploits these redundancies present in the 4D LF with the
proposal of efficient methods of 4D prediction and partition for the MRP codec. Two new modes and the prediction mode proposed by the authors in [17] are studied in this paper, each with its own characteristics in terms of the trade-off between complexity and compression efficiency. Although M-MRP was already introduced in [17], in the current paper its performance undergoes a more thorough evaluation, with updated test conditions, following the common test conditions of JPEG-Pleno (JPEG-Pleno CTC) standardisation initiative [19], and using a wider variety of datasets. M-MRP is presented alongside the new prediction modes so that their commonalities and differences are highlighted. The three modes studied in this paper are entitled the 4D-MRP, Dual-Tree 4D-MRP, and multiple-reference MRP (M-MRP) [17]. In the first, an hexadecac-tree partitioning structure is used to enable four-dimensional prediction and 4D blocks as coding units. In the second, the angular and spatial partitioning are separate, such that each 4D block can be partitioned into four blocks in either the angular dimensions \((t, s)\) or spatial dimensions \((v, u)\), rather than being partitioned in 16 blocks as in 4D-MRP. Finally, in M-MRP only 2D partitions are used, using the same 4D prediction of the other modes but losing the benefits of 4D partition. The main contributions of this paper can be summarised as follows:

- Development of a unified native 4D framework for lossless compression of light fields based on Minimum Rate Predictors.
- Design of two novel prediction modes for the native 4D lossless compression framework, namely 4D-MRP and DT-4D-MRP.
- Extensive evaluation and comparison of the three prediction modes in the context of the JPEG-Pleno CTC, amongst each other and with state-of-the-art encoders.
- Assessment of the three prediction modes in a medical context, using the Skin Lesion Light-fields (SKINL2) dataset [20].

The experimental assessment clearly ascertains the superior performance of the proposed methods in comparison with other state-of-the-art encoders, such as HEVC [21], VVC [22], and JPEG XL [23]. In fact, the presented methods surpass other encoders by up to 32%, in terms of bitrate savings.

The remainder of the paper is organised as follows. Section II-A presents the theory behind the 4D representation of LFs, Section II-B gives an overview of other lossless compression of LFs works, Section III describes the proposed encoders, in Section IV the experimental results are presented and analysed, and, finally, Section V discusses the outcomes of the paper.

II. BACKGROUND

A. 4D Light Field Representation

Light fields are sampled representations of the plenoptic function, which characterises the intensity \(I\) of light rays at every point in space \((x, y, z)\), travelling in the \((\theta, \phi)\) direction, with the wavelength \(\lambda\) at time instant \(t\) [24]. This is a 7D function represented by

\[
L = P(x, y, x, \theta, \phi, \lambda, t).
\]  

A 4D parametrisation of a light field using two planes is shown in Figure 1 where a light ray direction is defined by the coordinates of its intersection points with the \((v, u)\) and \((t, s)\) planes. Light rays coming from the same scene point with different directions pass through the same focal point in the \((v, u)\) plane and then are uniquely identified by the coordinates in the \((t, s)\) plane. In other words, for each point \((v, u)\) the camera captures the intensity of light rays with different directions \((t, s)\). In this context, \((t, s)\) represent the coordinates of sub-aperture images (SAI) in the 4D LF, and \((v, u)\) represent pixel coordinates in a SAI. The two planes represented in Figure 1 are discretised allowing only a finite number of viewpoints and light rays to be recorded.

Several processes for capturing LFs exist, as described by Zhou in [26]. For the purposes of this paper, single sensor LF cameras and array of cameras are considered. Considering the former, two main types of LF cameras can be found: the focused camera, also known as plenoptic 2.0 and the unfocused camera, also known as plenoptic 1.0. Several publicly available LF datasets can be found in [19] and [20].

However, in practical systems full representation of the plenoptic function is not necessary, thus a simplified version is normally used. This simplified version also requires more reasonable computational effort to sample due to inherent data reduction. Several different representations have been proposed in the literature each with distinct characteristics [25]. This paper employs the common 4D representation of LFs, which reduces the seven dimensions of the plenoptic function to four, based on the following three assumptions:

1) the wavelength \(\lambda\) can be simplified to the three RGB channels normally captured by cameras (that represent specific ranges of wavelength).

2) The transmission medium does not introduce attenuation, so the radiance along a light ray path remains constant. Therefore the scene information can be captured by projection onto any selected surface and the \(z\) dimension can be eliminated.

3) The scene is static during the acquisition interval thus, the time dimension \(t\) can be disregarded.

Under these assumptions the 7D plenoptic function can be simplified to a 4D function:

\[
L_{4D} = L(t, s, v, u).
\]

![Fig. 1. Simplified example of the 4D LF parametrisation in a plenoptic camera.](image-url)
Despite the differences in the acquisition processes, LFs can always be represented as a 4D data structure, comprising a set of SAIs with the same spatial resolution, each one representing a different viewpoint of the same visual scene. Usually these viewpoints are captured along the vertical and horizontal directions giving rise to disparity in both directions. The described 4D LF format is also adopted in JPEG-Pleno encoding framework and in their standard encoders [15], [27], which use the 4D LFs in the SAI domain [19].

B. Coding of Light Fields

Lossless compression techniques can be grouped according to LF representation format: lenslet sensor data, SAI or epipolar images (EPI). Considering the first category, encoders compress the raw ‘RGGB’ Bayer image obtained from the camera sensor, before the pre-processing steps, as in [28]. In this case, the metadata of the camera must be taken into account when calculating compression efficiency, as this information is required to extract images from the LF. In [29] the author proposes to split the image into four colour planes that are entropy analysed to discover the horizontal and vertical displacement between the micro images. Later, a sequential differential prediction process is performed, and the residue is encoded using a Lempel-Ziv-Markov chain algorithm [30]. The authors in [31] propose a different approach, where the nine closest causal micro-images are used to predict the current micro-image. This is achieved by designing a minimum description length optimal sparse predictor for each micro-image, where the non-zero coefficients of the predictor are selected by using a binary mask, which also needs to be transmitted. More recently, in [32] the authors propose a codec for plenoptic camera sensor images dubbed sparse relevant predictors, i.e.

The described 4D LF format is also adopted in JPEG-Pleno encoding framework and in their standard encoders [15], [27], which use the 4D LFs in the SAI domain [19].

Despite the previous methods being specifically tailored for lenslet formats, the major focus of LF lossless coding has been on compressing SAIs. In [37], [38], and [39] Helin et al. partition the views into regions, which are expanded from the central view using vertical and horizontal displacements that need to be transmitted. The central view is losslessly encoded with JPEG2000, the side views use the previously encoded ones as reference with a region specific sparse linear predictor. In [40] and [41] Schiopu et al. propose to use segmentation, after the predictive coding of the LF views, to improve the context modelling of the residuals for the entropy coder. Rizkallah et al. in [42] and [43] propose to use graph-based transforms for quasi-lossless compression of LFs.

Finally, few works are dedicated to the lossless compression of LFs in the EPI format. Mukati and Forchhammer propose to adapt the CALIC codec [44] to compress EPI images [45]. The CALIC GAP predictor is improved so that it can detect the EPI slope, thus the name ‘EPI slope based predictor’ (ESP), in order to appropriately predict the pixel intensities along each epipolar line. This proposal is shown to increase the compression performance of CALIC for LFs by 0.8 bit-per-pixel (bpp). This work is further expanded in [46], which increases the performance of the encoder by 5.3%.

The authors’ previous works [47] and [48] fall in all previous three categories, by comparing the performance of various state-of-the-art encoders on various LF data arrangements, with focus on Minimum Rate Predictors encoder (MRP) [49]. In these works various reversible colour transforms are tested to increase the lossless compression performance of such encoders.

Another work worth mentioning is [50], resulting from a JPEG-Pleno experiment intended to compare the lossless and near-lossless performance of upcoming JPEG-Pleno standard codecs, the WaSP and MuLE (which is currently lossy only) encoders to M-MRP [17]. MuLE and WaSP present lower bit-rate to obtain near-lossless compression, when compared with the lossless bit-rate of the M-MRP. However, when targeting lossless compression, the JPEG-Pleno standard codecs are unable to keep up with the M-MRP performance, in some cases achieving twice the bit-rate of M-MRP.

III. 4D PREDICTION MODES FOR MRP

A. The MRP coding algorithm

To cope with the content diversity of generic images, MRP uses an adaptive prediction scheme for each image. As for the prediction residue, the algorithm employs entropy coding based on context modelling. Such prediction residues are sorted into one of several predetermined groups, i.e. contexts, depending on their characteristics. The set of residues pertaining to a given group are later represented by a single generalised Gaussian probability density function [18] whose parameters differ from those of the other groups.

Mathematically, MRP is formulated as the search for linear predictors, i.e. classes, that minimise the number of bits used to encode the prediction errors and associated information. Classes are characterised by linear prediction models each with a set of coefficients. As described in [18], the information
where \( I(R) \) indicates the estimate of the total amount of information measured in bits, \( N \) is the number of groups associated with the context modelling, \( g_n \) are the pixels in the \( n^{th} \) group, \( \sigma_n \) is the variance of the prediction error \( e \) of the \( n^{th} \) group, \( e \) is Euler’s number, and \( a_n = \frac{\Delta e}{\sqrt{2\pi\sigma_n^2}} \), with \( \Delta e \) representing a sufficiently small quantisation step-size of \( e \). The objective of the encoding procedure is to minimise \( I(R) \), by properly determining the sets of linear prediction coefficients for the \( M \) classes, that are used to compute the prediction and associated residuals. In practical terms, the regions in Equation (3) are the partition blocks and the actual minimisation target is given by a cost function accounting for the prediction residuals and associated side information. The classes linear prediction models in MRP are adapted to each image characteristics, through the use of a variable block size partitioning scheme in the iterative minimisation of the cost function. In general, the MRP algorithm processing flow can be represented by the block diagram shown in Figure 2, which is further described by Algorithm 1 where each colour represents the same operation in both the figure and the algorithm. The modules include the three main stages, Fixed Block-Size, Variable Block-Size, and Arithmetic Coding.

Prior to the actual encoding, a pre-processing step, consisting in applying a Reversible Colour Transform (RCT) and an histogram packing step, is applied to the LF. The RCT, which was first combined with MRP in [48], aims to lower the RGB inter-component correlation to improve the coding performance, the transform is defined in Equation (4) where \( [\_] \) represents the floor operation. The histogram packing, first introduced to MRP in [49], is used to reduce the sparseness of the LF histogram, improving the performance of lossless prediction based encoders, provided that it produces a LF
image with lower total variation than the original LF.

\[
Y = \left\lfloor \frac{R + 2G + B}{4} \right\rfloor \\
C_u = B - G \\
C_v = R - G
\]  \Rightarrow \begin{cases} 
R = C_u + G \\
G = Y - \left\lfloor \frac{C_u + C_v}{4} \right\rfloor \\
B = C_u - G 
\end{cases} \quad (4)

In the Fixed Block-Size stage blocks of 8 × 8 pixels are used to calculate the coefficients of the \( m \) linear prediction models that define the corresponding classes. First the blocks are sorted according to their pixel variance and distributed by the classes in an increasing variance order to initialize the algorithm. For each class \( m \), the coefficients \((a_{m_i})\) of the corresponding linear prediction model are determined in module Design Classes, by solving a set of Yule-Walker equations, which uses the pixels of all blocks associated with that class. Then, in Optimise Groups the thresholds for the quantisation of the prediction residuals context, represented by \( C \), are computed. The quantisation of the value \( C \) generates 16 different contexts, given by 15 thresholds that are calculated on a class basis. This means that all the blocks of a given class are used to compute these thresholds. Due to the dynamic nature of the thresholds, these are explicitly transmitted in the bitstream. Finally, in Optimise Classification the 8 × 8 blocks are reclassified into the newly computed classes linear prediction models, with the objective of minimising the optimisation target. Each class is tested in order to find the one that minimises the block prediction residuals cost. Instead of directly transmitting the actual class index for each block, the MRP uses a lookup table that is updated as each block is processed by using a move-to-front method. The class index associated with the upper, left, and upper right neighbour blocks are placed at the top of the lookup table, in this order. In this manner, as neighbour blocks are correlated, the indices to transmit should generally have small values, which leads to a more efficient compression of these indices (i.e., different classes can be represented by the same lookup table index). Additionally, due to the optimisation process, classes of neighbouring blocks are favoured, especially in flat areas. In this first optimisation loop, the minimisation target is the prediction residuals encoding cost estimated by:

\[
B_r = \sum_{n_0} L(e|\hat{s}(0), n), \quad (5)
\]

where \( L(e|\hat{s}(0), n) \) is the encoding cost of the prediction error, which depends on the probability density estimate of the \( n \)-th group, assumed to conform with a Gaussian distribution. By minimizing the cost \( B_r \), that is, the encoding costs of the prediction errors, the method implicitly optimizes the information \( I(R) \) in a given region (or blocks), therefore minimizing Equation 4.

The Variable Block-Size stage refines the operations of the previous optimisation loop, by taking into account the cost of all signalling flags, using variable block size and fitting the probability models. In addition to the operations of the previous loop, Optimise Groups updates the shape parameter of generalised Gaussian probability density functions to get a good fit for the prediction error of each group. In Optimise Classification, the classes linear prediction models designed in the previous stage are used to optimise the block classification resorting to variable block sizes, partitioned in a quadtree fashion. The block sizes range from 32 × 32 pixels down to 2 × 2 pixels, for quadtree levels from 4 down to 0, respectively. The optimal partition is selected so that at each level, i.e. block size, the minimum between the current block cost \( (J_t) \) and the sum of the costs of the sub-blocks \( (J_2) \) is selected. The target cost function of the second loop is defined as a sum of several costs expressed in bits:

\[
J = B_a + B_m + B_i + B_r, \quad (6)
\]

where \( B_a \), \( B_m \), and \( B_i \) are the encoding costs of the prediction coefficients, class selection, and context modelling threshold values, respectively, and \( B_r \) is defined in Equation 5.

The resulting information from these two optimisation loops is then encoded using an arithmetic encoder [51]. More details on the LF encoding based on the MRP algorithm can be found in [47], [48].

As mentioned before, the focus of this work is the development of new native 4D frameworks, i.e., 4D prediction modes for MRP, to better exploit the pixel redundancies in the 4D space of LF representation \( \bar{L}(t, s, v, u) \) described in Section II-A. Such redundancies are evidenced in Figure 3 as discussed in the same section and confirmed by Figure 4 which shows the result of a zeroth-order angular prediction (i.e., prediction residue), obtained by subtracting the average of the top and left neighbouring SAIs from the current one, the top, and left ones. This simple experiment demonstrates that the energy of prediction residuals is quite low, indicating that more advanced prediction modes, which fully exploit the 4D space, are expected to achieve significant improvement in coding efficiency.

B. 4D-MRP

The four-dimensional Minimum Rate Predictor (4D-MRP), stands on two main pillars: four dimensional prediction and partition of the LF into 4D blocks for the selection of the class. The motivation for using 4D blocks is supported by the fact that, in general, the same class is selected for co-located blocks in neighbouring SAIs. Figure 5 displays this behaviour, where each class is represented by a different colour. It can be seen
which is represented in Figure 6, where four causal SAIs
causal vicinity defines the SAIs that can be used as references,
i.e. current pixel, thus the need to use a causal vicinity. In this
is pixel-based, not all SAI pixels might be available for the
number of reference pixels in the MRP prediction. Since MRP
4D-MRP, causal neighbouring SAIs are used to extend the
is avoided.

A 2D one, signalling each 2D block partitioning in each SAI
indicates that bitrate savings can be achieved in the signalling
cost incurred by the quadtree. By using a 4D block instead of
a 2D one, signalling each 2D block partitioning in each SAI
is avoided.

Therefore, in the novel prediction mode proposed in the
4D-MRP, causal neighbouring SAIs are used to extend the
number of reference pixels in the MRP prediction. Since MRP
is pixel-based, not all SAI pixels might be available for the
current pixel, thus the need to use a causal vicinity. In this
context, causality is defined by the encoding order of the SAIs,
i.e., causal SAIs are the ones that were already encoded. The
causal vicinity defines the SAIs that can be used as references,
which is represented in Figure 6, where four causal SAI
neighbours are used, namely: left (l), left diagonal (tl), top
(t), and right diagonal (tr). The grey line divides the causal
pixels. Equation [7] shows how the prediction is computed for
pixel \( p_0 \), as a linear combination of the neighbouring pixels:

\[
\hat{s}(0) = \sum_{k=1}^{K_c} a_m(c, k) \cdot s(c, k) + \sum_{k=1}^{K_l} a_m(l, k) \cdot s(l, k) \\
+ \sum_{k=1}^{K_{tl}} a_m(tl, k) \cdot s(tl, k) + \sum_{k=1}^{K_t} a_m(t, k) \cdot s(t, k) \\
+ \sum_{k=1}^{K_{tr}} a_m(tr, k) \cdot s(tr, k),
\]

(7)

where \( a_m(i, k), i = \{c, l, tl, t, tr\} \) represent the prediction
model weights associated with pixels in the current SAI (c)
and the four causal neighbouring references, respectively. For
the \( n \)-th class, \( s(i, p_0), i = \{c, l, tl, t, tr\} \) are the pixel values
in the current and reference SAIs at position \( k \) of the prediction
support region, and \( K_i, i = \{c, l, tl, t, tr\} \) are the sizes of each
prediction support region. The prediction support regions need
not be the same size or shape.

The context \( C \) associated with the \( p_0 \) pixel prediction error
is calculated as the sum of the reference pixels prediction
errors, \( e_{i,k} \), weighted by the inverse of the distances to these
pixels, given by:

\[
C = \sum_{k=1}^{K_c} \frac{1}{\delta_k} e_c + \sum_{k=1}^{K_l} \frac{1}{\delta_k} e_l + \sum_{k=1}^{K_{tl}} \frac{1}{\delta_k} e_{tl} \\
+ \sum_{k=1}^{K_t} \frac{1}{\delta_k} e_t + \sum_{k=1}^{K_{tr}} \frac{1}{\delta_k} e_{tr},
\]

(8)

where \( e_i = |s(i, k) - \hat{s}(i, k)| \) and \( \delta_k, i = \{c, l, tl, t, tr\} \)
are weighting factors proportional to the Euclidean distance
between the current pixel (\( p_0 \)) and their references (\( p_k \)). The
\( \delta_k \) weight is defined as \( \delta_k = \sqrt{d_c(k)^2 + d_{l}(k)^2 + d^2} \),
where \( d \) represents the distance between SAIs and is set to 1 for
all references except the current one, for which it is 0. The
resulting \( C \) undergoes a quantisation step, using the thresholds
described in Section III-A and is used to select the adaptive conditional probability models.

In MRP, a set of prediction coefficients, \( a_m(i, k) \), are determined for each class independently. Then each pixel of the current SAI is assigned a class corresponding to the predictor that minimises Equation 6. By using 4D blocks, the MRP quadtree partition is expanded into a hexadecatree, where each block is divided in all four dimensions simultaneously, resulting in 16 sub-blocks in the 4D space. Let a block be represented by \( L_B(t, s, v, u, w) \), with \( t \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < T \} \), \( s \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < S \} \), \( v \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < V \} \), and \( u \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < U \} \), in the block coordinates, then the hexadecatree partition can be represented by Equation 9:

\[
P_{\text{hex}}(L_B(t, s, v, u)) = \{ L_B(t, s, v, u), \forall i \in \{ 0, \ldots, 15 \} \},
\]

with

\[
\begin{align*}
t_i &= a + T \times (1 + (-1)^{i/8} + 1)/4, \\
s_i &= b + S \times (1 + (-1)^{i/4} + 1)/4, \\
v_i &= c + V \times (1 + (-1)^{i/2} + 1)/4, \\
u_i &= d + U \times (1 + (-1)^{i+1} + 1)/4,
\end{align*}
\]

\[\forall a \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < T/2 \}, \]
\[\forall b \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < S/2 \}, \]
\[\forall c \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < V/2 \}, \]
\[\forall d \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < U/2 \}
\]

where \( T, S, V \) and \( U \) represent the dimensions of the 4D block.

Figure 7 represents the tree structure partition and Figure 8 shows a visual representation of the 4D block partition. In Figure 8, the angular dimensions \((t, s)\) are partitioned by the green lines and spatial dimensions \((v, u)\) by the blue lines, the index of each 4D block is shown in white circles. As pointed out before, the hexadecatree partitioning is expected to result in a larger number of pixels being classified with lower signalling cost, thus increasing compression efficiency.

The hexadecatree partition and block classification optimisation is performed at the Optimize Classification stage, shown in Figure 2. This optimization aims at minimising both side information, represented by \( B_m \) in Equation 6 and the prediction error cost. \( B_m \) can be further represented by:

\[
B_m = B_{\text{flags}} + B_{\text{class}},
\]  

where \( B_{\text{flags}} \) given by Equation 11 represents the cost of selecting the partition flags and \( B_{\text{class}} \) the cost of selecting a class for each resulting block.

\[
B_{\text{flags}} = \begin{cases} 
- \log_2(P_{\text{ctx}}), & \text{not partitioned} \\
- \log_2(1 - P_{\text{ctx}}), & \text{partitioned}
\end{cases}
\]  

In Equation 11, \( P_{\text{ctx}} \) represents an approximation of the probability of the blocks in a given context not being partitioned, given by:

\[
P_{\text{ctx}} = \arg\min_{p} \left\{ - \log_2(p) \times \#ctx_0 - \log_2(1 - p) \times \#ctx_1 \right\},
\]

with \( p \in \{ 0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95 \} \),

where \( \#ctx_0 \) and \( \#ctx_1 \) represent the number of occurrences of the 0 and 1 partition flags, respectively, in each context, \( i.e. \) the number of times each block is either not-partitioned or partitioned in each given context. \( p \) represents the probability that minimises the cost of selecting a flag in each context. This probability is selected from a discrete set to minimise the cost of transmitting the real probability value, therefore only a single index needs to be sent to the decoder. The quantified probability calculation of Equation 12 follows that used in MRP, which was first introduced with the variable block size in [13].

Finally, the context calculation is determined at each level by the number neighbouring blocks which were partitioned at said level:

\[
C_{\text{flags}}(\text{level}, b) = \sum_{b \in \mathcal{N}} q_{\text{flags}}(\text{level}, b),
\]

where \( b \) represents a block, \( q_{\text{flags}} \) the partition flag for block \( b \) with the size given by \( \text{level} \), 0 and 1 for non-partitioned
and partitioned blocks, respectively, and \( N \) represents the neighbouring blocks on the same level as \( b \). As 4D-MRP uses 4 levels for the block sizes and in the hexadecatree 5 neighbours are being considered, there are a total of 24 different contexts.

In the Design Predictors stage of the encoding process, fixed size blocks are used to determine the linear predictor coefficients of each class. These operations are affected by the block size, as is the coding performance, through the calculation of the coefficients \( a_{m_i}(k) \), which decrease with the larger number of pixels. Experiments showed that the \( 8 \times 8 \times 8 \times 8 \) pixels fixed block-size, analogous to what is used in MRP, was too large for the 4D space, which resulted in blocks of \( 8^4 = 4096 \) pixels total. The same experiments showed that blocks of \( 4^4 = 256 \) pixels were a good compromise in terms of coding ratio. Thus in this work, when 4D blocks are used in the Fixed Block-Size part, the size is set to \( 4 \times 4 \times 4 \times 4 \). In the Variable Block-Size part the top level block size is set to \( 32 \times 32 \times 32 \times 32 \) pixels.

C. Dual-Tree 4D-MRP (DT-4D-MRP)

1) Dual-Tree Partitioning: This partitioning mode is intended to improve the problem created by hexadecatrees, which tend to use very fine partitioning with a high number of \( 2 \times 2 \times 2 \times 2 \) blocks (the smallest possible size), albeit still achieving better performance than MRP, as shall be seen in Section IV. As expected, using smaller block sizes requires more signalling bits to encode the partition tree. An analysis of the costs and benefits of the partitioning alternatives leads to the hypothesis that independently partitioning the spatial and angular dimensions could lead to further coding gains. This is supported by different sampling present on the spatial and angular dimensions at the acquisition.

In this mode, the block partitioning algorithm was changed to decouple the angular from the spatial partitioning. Therefore, rather than partitioning in 16 blocks as in 4D-MRP, each 4D block can be partitioned either into four blocks in the angular dimensions \((t, s)\), into four blocks in the spatial dimensions \((v, u)\), or not partitioned at all. Let a block be represented by \( L_B(t, s, v, u) \), with \( t \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < T \} \), \( s \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < S \} \), \( v \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < V \} \), and \( u \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < U \} \), in the block coordinates. Then the dual-tree partition can be defined by Equations (14) and (15) for the spatial and angular partitions respectively.

\[
P_{\text{DT}_3}(L_B(t, s, v, u)) = \{ L_B(t, s, v, u), \forall i \in \{0, \ldots, 3\} \},
\]

with

\[
\begin{align*}
&\forall t \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < T \}, \\
&\forall s \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < S \}, \\
&\quad v = a + V \times (1 + (-1)\{[v/2]+1\}/4, \\
&\quad u = b + U \times (1 + (-1)\{[u/2]+1\}/4, \\
&\quad \forall a \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < V/2 \}, \\
&\quad \forall b \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < U/2 \},
\end{align*}
\]

\[
P_{\text{DT}_A}(L_B(t, s, v, u)) = \{ L_B(t, s, v, u), \forall i \in \{0, \ldots, 3\} \},
\]

with

\[
\begin{align*}
&\forall t \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < T \}, \\
&\forall s \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < S \}, \\
&\quad v = a + V \times (1 + (-1)\{[v/2]+1\}/4, \\
&\quad u = b + U \times (1 + (-1)\{[u/2]+1\}/4, \\
&\quad \forall a \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < V/2 \}, \\
&\quad \forall b \in \{ \gamma \in \mathbb{N} | 0 \leq \gamma < U/2 \},
\end{align*}
\]

This structure of 4D blocks is represented in Figure 9, where ‘N’ represents the non-partitioned blocks, ‘S’ the spatial partition, and ‘A’ the angular partition. Figures 10a and 10b show visual representations of the separate quadtrees for spatial and angular dimensions, respectively.

2) Arithmetic Coding of the Partition Tree Signalling: In this new tree structure the decisions are no longer binary but ternary, since there are three options for partitioning each block (i.e., spatial, angular, none). In DT-4D-MRP the cost of selecting a given partition mode was redefined to reflect the new signalling structure:

\[
B_{p_{\text{flags}}} = \begin{cases} 
-\log_2(P_N), & \text{if flag = ‘N’} \\
-\log_2(P_{S|\bar{N}} \cdot P_{\bar{N}}), & \text{if flag = ‘S’}, \\
-\log_2((1 - P_{S|\bar{N}}) \cdot P_{\bar{N}}), & \text{if flag = ‘A’}
\end{cases}
\]

where ‘N’, ‘S’, and ‘A’ represent the partition flags of a block, indicating whether a block was not partitioned, partitioned in the spatial dimensions, or partitioned in the angular dimension, respectively, as shown in Figure 9. \( P_N \) represents the probability of the ‘N’ flag, \( P_{S|\bar{N}} \) the complementary probability of \( P_N \), and \( P_S \) the probability of the ‘S’ flag given \( \bar{N} \), resulting from:

\[
P_N = \arg\min_p \{-\log_2(p) \times ctx_N - \log_2(1-p) \times (ctx_S + ctx_A)\},
\]

\[
P_{S|\bar{N}} = \arg\min_p \{-\log_2(p) \times ctx_S - \log_2(1-p) \times ctx_A\},
\]

(17)

where \( ctx_i \) represents context_\text{DT}(i), given by Equation (18) and \( p \in \{0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95\} \).

For the DT-4D-MRP the number of possible contexts was reduced, and the computation of the context was replaced by the number of times each flag (‘N’, ‘S’, or ‘A’) is chosen in the tree:

\[
C_{q_{\text{flags}}\text{DT}}(F) = \sum_{b \in \text{tree}} \delta_{\text{flag}(b), F},
\]

(18)
where $F$ represents the flag for which the context is being determined, $b$ represents each block in the tree, flag($b$) is the partition flag of each block $b$ and $\delta$ is the Kronecker delta.

These operations are represented by Algorithm 2. As in the 4D-MRP case, a table of probabilities (treeprob) is used to facilitate the encoding of the partition flags probability, as the index is more efficiently encoded than the actual probability value. The optimal index $o_{index}$ corresponds to the probability $p$ that minimises the expression in Line 6 and Line 18 of the algorithm. The cost of each flag is represented by the variable: cost = $-\log(p)$. This calculation is performed in two steps, initially the cost of the ‘N’ flag is determined, by grouping the probability of the remaining flags, ‘S’ and ‘A’. In the second step, only ‘S’ and ‘A’ are taken into account, and their relative probabilities are determined. For the actual cost, these probabilities are normalised by multiplying them by a factor of $1-p_{ex}$, such that $\sum_{j=N,S,A} P_j = 1$. These values are used to set the probability models for the arithmetic coding of the tree partition, and the previously mentioned index is explicitly transmitted in the bitstream. Each flag cost is used in the optimisation of the partition tree. On the decoder side, the partition flags are decoded using the transmitted probability indices, without the need of recalculating the contexts of Equation 18.

The remaining steps in the arithmetic coding of the partition flags and class selection are kept unchanged. Essentially, starting with the first block, the partition flags are encoded until a block is no longer partitioned, then the selected class is encoded in the bitstream.

### Algorithm 2 Calculation of the partition flags cost.

1: cost = inf
2: treeprob = [0.05, 0.2, 0.35, 0.5, 0.65, 0.8, 0.95]
3: #Determine cost of symbol ‘N’
4: for $i$ = 0 to 6 do
5: $p = \text{treeprob}[i]$
6: $c = -\log(p)\cdot \text{context}[‘N’] - \log(1-p)\cdot \text{context}[‘S’] + \text{context}[‘A’]$
7: if $c < \text{cost}$ then
8: cost = $c$
9: $o_{index} = i$
10: $p_N = \text{treeprob}[o_{index}]$
11: flagcost[‘N’] = $-\log_2(p_N)$
12: cost = inf
13: #Determine cost of symbols ‘S’ and ‘A’
14: for $i$ = 0 to 6 do
15: $p = \text{treeprob}[i]$
16: $c = -\log(p)\cdot \text{context}[‘S’] - \log(1-p)\cdot \text{context}[‘A’]$
17: if $c < \text{cost}$ then
18: cost = $c$
19: $o_{index} = i$
20: $p = \text{treeprob}[o_{index}]$
21: flagcost[‘S’] = $-\log_2(p\cdot (1-p_N))$
22: flagcost[‘A’] = $-\log_2((1-p)\cdot (1-p_N))$

Another alternative to deal with the large overhead of hexadecimal tree partitioning signalling is to abandon 4D partitioning and perform only 2D partitions. This is done by treating the LF as a single 2D image for the partition, which results in a better spatial partition, but forsakes the potential benefits of the 4D partition presented in 4D-MRP. The multiple reference MRP (M-MRP), which, as mentioned before, was described in [17], combines the 4D prediction described in the Section III (represented in Figure 6) with the conventional 2D quadtree partition, represented in Figure 11 and Figure 12. Let a 2D block be represented by $L_B(v, u)$, with $v \in \{\gamma \in \mathbb{N} | 0 \leq \gamma < V\}$, and $u \in \{\gamma \in \mathbb{N} | 0 \leq \gamma < U\}$, in the block coordinates,
then the M-MRP partition can be represented by:

\[
P_{DT_s}(L_B(v, u)) = \{L_{B_i}(v, u), \forall i \in \{0, \ldots, 3\}\},
\]

\[
\begin{align*}
v &= a + V \times (1 + (-1)^{|i/2|+1})/4, \\
u &= b + U \times (1 + (-1)^{|i+1|}/4,
\end{align*}
\]

\[
\forall a \in \{\gamma \in \mathbb{N} | 0 \leq \gamma < V/2\}, \\
\forall b \in \{\gamma \in \mathbb{N} | 0 \leq \gamma < U/2\}
\]

Regarding the partition structure, as this is similar to 4D-MRP, in particular, and to MRP, in general, no further modifications to the signalling structure are needed.

IV. EXPERIMENTAL EVALUATION

The experimental evaluation follows the common test conditions of JPEG-Pleno (JPEG-Pleno CTC) [19], with minor changes to adapt to lossless compression. The experiments use two of the JPEG-Pleno CTC datasets, namely the Lenslet Lytro Illum Camera dataset from EPFL [59] and the Synthetic HCI HDCA dataset from HCI [52]. Additionally, to expand the range of these experiments a dataset of skin lesions, Skin Lesion Light-fields (SKINL2) [20], was also used. The SKINL2 dataset was acquired with a Raytrix R42 camera in a clinical setting in the context of dermatological research on skin lesions using LFs. The images used in this work were also submitted to the JPEG-Pleno standardisation initiative [54]. Figure 13 shows a SAI of one image of each of these datasets, and Table I lists their characteristics.

The JPEG-Pleno toolchain uses the LF Toolbox [28] to decode the raw lenslet files from the Lytro dataset, i.e. the EPFL dataset, and convert them to 4D LFs. The Colour Correction step of the pre-processing chain expands the colour range which is detrimental to the coding efficiency, therefore, as the compression process is lossless, this mentioned step is omitted before encoding and if necessary performed at the decoder side.

The performances of the proposed encoding methods were compared with those of other state-of-the-art encoders like MRP, JPEG-LS [57], [58], CALIC [44], [56], JPEG XL [23], HEVC [21], [59], and VVC [22], [60]. To ensure the uniformity of the experiments, the RCT is also applied to the input of the state-of-the-art encoders. The state-of-the-art encoders implementations and parametrisation are shown in Table II.

A. Comparison of proposals and state-of-the-art

The experimental results in Table III show the coding efficiency, measured in bits-per-pixel (bpp), i.e.

\[
bp p = \frac{B_{C_1} + B_{C_2} + B_{C_3}}{T \times S \times V \times U}
\]

where \(B_{C_i}\) is the number of bits of the compressed colour component \(C_i\) (e.g. YUV) of the image, \(T \times S\) represents the number of SAs, and \(V \times U\) the number of pixels of a SAI.

The three modes described in Section III 4D-MRP, DT-4D-MRP, and M-MRP, have been designed to exploit the unique characteristics of light-field images. Their coding efficiencies were evaluated by encoding all the images listed before. The results are analysed and discussed next.

1) Experiments using EPFL and HCI datasets: The results in Table III show that MRP surpasses standard codecs like JPEG, HEVC, or VVC. However, neither of these standard encoders, including MRP, is adapted to exploit the particular characteristics of LF representation data. Comparison of the proposed methods with the JPEG-LS, CALIC, JPEG XL, HEVC (HM), or VVC (VTM) coders, demonstrates the superiority of the former over the latter. Indeed the proposed 4D-MRP and DT-4D-MRP schemes surpass the compression

\[\text{Available at http://on.ipleiria.pt/plenoisla}\]
TABLE II
ENCODING CONFIGURATIONS OF THE STATE-OF-THE-ART ENCODERS USED IN THIS WORK.

| Encoder | Software | Input type | Configuration |
|---------|----------|------------|---------------|
| HEVC    | HM reference software v16.12, including the Format Range Extension (RExt) | PVS using spiral scan | - Main RExt profile (Profile=main-RExt) 
- Lossless cost mode 
- QP 0 (zero) 
- Transform quantization bypass (TransquantBypassEnable=1) 
- 32 intra period (IntraPeriod=32) 
- 16 GOP size (GOPSsize=16) 
- CRA intra random access point (open GOP) (DecodingRefreshType=1) 
- TZ Search motion estimation (FastSearch=1) |
| VVC     | VTM reference software v10.0 | PVS using spiral scan | - Using configuration file: Random Access GOP32 
- Using lossless configuration files 
- QP 0 (zero) |
| JPEG-LS | JPEG-LS reference software | Array of SAIs | - Standard configuration (-ls 1) 
- YCbCr colour transformation bypass active (-c) |
| CALIC   | Martin Briano’s implementation | Array of SAIs | - Lossless mode (-d=0) 
- 2D CALIC (-m 1) |
| JPEG XL | JPEG XL reference software v0.7.0 | Array of SAIs | - Lossless mode (-d 0) 
- Maximum effort (-e 9) 
- Parallel computations disabled (-num_threads=0) 
- Maximum number of classes dependent on image size 
- Bidirectional prediction with GOPSize = 8 (-B -G 7) 
- Histogram packing and variable block size prediction enabled |
| MRP     | Author’s software from MRP by 1.00 to 1.09 bpp on average, respectively. This represents roughly 10% bit-rate savings. Considering the most recent state-of-the-art image and video coding standard, VVC, the proposed methods present bit-rate savings of 15%, on average, surpassing VVC (VTM) by 1.66 to 1.75 bpp, for 4D-MRP and DT-4D-MRP, respectively. To the best of the authors’ knowledge DT-4D-MRP achieves the overall highest compression efficiency in the universe of state-of-the-art lossless LF encoders. Considering the intra only encoders, JPEG-LS, CALIC, and JPEG XL, it is clear that their performance is quite worse than the remaining encoders. This can be explained by the lack of tools to exploit inter SAI redundancy that the all the other codecs have in various ways.

Regarding the individual evaluation of the three proposed methods, it can be seen that the DT-4D-MRP achieves the highest compression efficiency, both on average and for each image, by 0.6 to 0.9%, when compared to the 4D-MRP and M-MRP, respectively. The dual quadtree prediction provides better adaptation of partitions to the image data, which in turn results in higher compression efficiency. The 4D-MRP encoder presents the lower coding efficiency of the three, which is explained by the highly granular partition used by the encoder. This is due to the simultaneous partitioning in all dimensions, which might not gather so much redundancies as possibly expected, also resulting in smaller blocks and more side information. M-MRP, which uses a 2D instead of 4D
partitions, achieves an intermediate coding efficiency when compared with 4D-MRP and DT-4D-MRP. This is due to the fact that 2D partitions can use blocks that include pixels from multiples SAIIs and also because larger blocks might be used, as splitting in the four-dimensions at once is not mandatory.

2) Experiments using SKINL2 dataset: Further coding experiments using the SKINL2 dataset revealed that performance results are, generally, consistent with those observed with the non-medical images. The results shown in Table IV confirm the better performance of the the proposed methods, which achieve higher compression efficiency than the remaining state-of-the-art encoders, surpassing MRP by 0.85 to 1.02 bpp, for 4D-MRP and M-MRP, on average, which represent bit-rate savings larger than 11%. In comparison with the VVC (VTM), the proposed methods present gains of 1.46 to 1.63 bpp, on average, for 4D-MRP and M-MRP, respectively, which represent up to 19% of bit-rate savings. These results show that the proposed methods achieve a higher compression efficiency for the skin lesions dataset, when compared with the remaining encoders. This is a noteworthy conclusion, as in many application scenarios medical images are required to be losslessly compressed. Curiously, unlike what was shown in Table III, while JPEG-LS and CALIC still present the worst compression performance, JPEG XL is able to compete with the remaining encoders, despite not being able to exploit inter SAI redundancies. In particular, JPEG XL is only surpassed by the methods of the proposed unified native 4D framework, with savings ranging from 0.71 bpp to 0.88 bpp, or 9.1% to 10.9%. This might be explained by the group splitting of JPEG XL, which divides larger images into sub images of $256 \times 256$ pixels to be encoded separately, and the characteristics of the SKINL2 dataset.

Regarding the comparison between the three proposed methods, it can be observed that DT-4D-MRP and M-MRP achieve similar compression efficiency, with a slight advantage for M-MRP. For this dataset (SKINL2) the bit-rate savings of M-MRP are 2% and 0.43% when compared with 4D-MRP and DT-4D-MRP, respectively. This shows that there is a relative loss of efficiency of 4D-MRP and DT-4D-MRP for the current dataset, when compared with the results of Table III. This might be explained by the larger resolution of the images in the SKINL2 dataset, that seems to favour the M-MRP algorithm.

Overall the experimental evaluation demonstrates, in a consistent manner, that using both 4D prediction and 4D partition to encode LFs in MRP, leads to the higher coding efficiency of DT-4D-MRP.

### Table III

**Compression results (in bpp) for EPFL and HCI datasets.**

| Type     | Light Fields | JPEG-LS | CALIC | JPEG XL | HM | VVC    | MRP [49] | 4D-MRP | M-MRP [17] | DT-4D-MRP |
|----------|--------------|---------|-------|---------|----|--------|----------|--------|------------|-----------|
| LL       | Bikes        | 16.42   | 15.94 | 16.07   | 13.69 | 13.72  | 12.73    | 11.43  | 11.47      | 11.34     |
|          | Danger_de_Mort | 16.18   | 15.69 | 15.77   | 13.03 | 12.97  | 11.89    | 10.80  | 10.80      | 10.71     |
|          | Fountain_and_Vincent_2 | 16.66   | 16.23 | 16.32   | 14.43 | 14.52  | 13.28    | 12.09  | 12.06      | 11.99     |
|          | Stone_Pillars_Outside | 16.77   | 16.28 | 16.33   | 13.44 | 13.38  | 12.36    | 11.29  | 11.20      | 11.17     |
| HDCA     | Greek        | 7.25    | 7.17  | 6.54    | 5.55 | 5.36   | 5.40     | 4.82   | 4.78       | 4.78      |
|          | Sideboard    | 12.19   | 12.17 | 11.04   | 8.17 | 8.09   | 8.43     | 7.66   | 7.62       | 7.58      |
| Average  |              | 14.24   | 13.91 | 13.68   | 11.39 | 11.34  | 10.68    | 9.68   | 9.66       | 9.59      |

B. Analysis of computational complexity

Besides coding efficiency, the computational complexity is an important benchmarking aspect, although not always primary. The complexity results pertaining to the different encoders, evaluated by encoding the EPFL and HCI dataset images, are presented in Table V. These results were obtained on a server running Ubuntu Server 20.04.1 LTS, equipped with an Intel Xeon(R) Silver 4114@2.20GHz CPU, and 192GB of DDR4@2666MHz RAM. The results show that the DT-4D-MRP has the worst performance in terms of complexity, when compared with the other encoders, taking on average 78 times longer than MRP to encode the same images. As stated in Section III-C this high computational complexity is due to the large number of partitioning options that need to be tested in the dual quadtree. The 4D-MRP and M-MRP proposals have computational complexities of the same magnitude, about 4 and 7 times that of MRP respectively. However they have a smaller compression performance than that of DT-4D-MRP. Despite VVC presenting about the same complexity as the 4D-MRP and M-MRP encoders, its compression performance is much lower than that of these encoders. Its performance is on pair with that of HEVC, despite being an older standard. This might be explained due to the developing effort of VVC having been so far focused on lossy compression performance with smaller improvements to the lossless modes.

The three prediction modes described in Section III increase the compression efficiency of MRP encoding and to the best of the authors’ knowledge, DT-4D-MRP presents the best lossless performance for LF images. While its computational complexity might be a concern, some adjustments can be made to deal with it. The authors have shown in [61] that the complexity of M-MRP can be lowered, while still maintaining better compression efficiency than other state-of-that-art encoders. Additionally, none of the computer code of the proposed methods was optimised for computational efficiency, which leaves room for improvements. In the case of 4D-MRP and M-MRP, while they present lower performance than DT-4D-MRP, still have competitive coding ratios and can be regarded as a good compromise in terms of rate-complexity trade-off. Most of the computational complexity of the proposed framework lays at the encoder side, with the three methods having about the same computation times for the decoder. Given these considerations, the DT-4D-MRP might be a good candidate for offline compression. One application is related to large databases, such as in medical facilities, where the
TABLE IV

| Light Fields      | JPEG-LS | CALIC | JPEG XL | HM | VVC | MRP [49] | 4D-MRP | M-MRP [17] | DT-4D-MRP |
|-------------------|---------|-------|---------|----|-----|----------|--------|------------|-----------|
| Img1              | 13.39   | 12.63 | 7.76    | 8.65| 8.17| 8.06     | 7.01   | **6.79**   | 6.84      |
| Img2              | 13.59   | 12.79 | 7.90    | 9.10| 8.74| 7.81     | 7.10   | **6.98**   | **6.97**  |
| Img3              | 13.61   | 12.82 | 7.80    | 9.18| 8.79| 8.00     | 7.23   | **7.07**   | 7.09      |
| Average           | **13.53** | 12.75 | 7.82    | 8.98| 8.57| 7.96     | 7.11   | **6.94**   | 6.97      |

TABLE V

| Light Fields | Type  | JPEG-LS | CALIC | JPEG XL | HM | VVC | MRP [49] | 4D-MRP | M-MRP [17] | DT-4D-MRP |
|--------------|-------|---------|-------|---------|----|-----|----------|--------|------------|-----------|
| Bikes        | LL    | 1.5E-3  | 1.0E-2| 0.2     | 0.2| 3.6 | 1.0      | 3.5    | 6.4        | 91.9      |
| Danger_de_Mort|       | 2.0E-3  | 1.4E-2| 0.2     | 0.3| 4.4 | 1.0      | 4.7    | 9.3        | 90.2      |
| Fountain_and_Vincent_2|   | 1.9E-3  | 1.1E-2| 0.2     | 0.3| 4.6 | 1.0      | 3.7    | 8.8        | 122.5     |
| Stone_Pillars_Outside|   | 1.7E-3  | 1.1E-2| 0.2     | 0.3| 4.5 | 1.0      | 3.7    | 8.0        | 89.0      |
| Greek        | HDCA  | 8.9E-4  | 6.2E-3| 0.1     | 0.1| 3.0 | 1.0      | 4.9    | 6.6        | 35.1      |
| Sideboard    |       | 1.0E-3  | 6.4E-3| 0.1     | 0.2| 2.6 | 1.0      | 4.3    | 4.1        | 44.4      |
| Average      |       | **1.5E-3** | **9.8E-3** | **0.2** | **0.2** | **3.8** | **1.0** | **4.1** | **7.1** | **78.9** |

Compression efficiency is more important than the computation times. In this case, any amount of extra compression can lead to a more efficient use of the available storage capacity. On the other hand, when computational efficiency is a concern, for instance for domestic users, one of the other prediction modes can be chosen.

V. CONCLUSION AND FUTURE WORK

A. Conclusion

In this paper, three 4D prediction and partition modes for LF lossless encoding are studied. These allow the encoder to use up to four neighbouring SAIs as prediction references, which leads to better prediction performance and increases the coding efficiency of the MRP encoding, by exploiting the characteristics of LFs. The three types of prediction used to select the best predictors use 4D partitions with (i) an hexadeca tree (4D-MRP), (ii) a 4D partition using a quadtree for the spatial dimensions and other for the angular dimensions (DT-4D-MRP), and (iii) a conventional 2D partition (M-MRP) with multiple references, previously described in [17]. The 4D-MRP and M-MRP prediction modes present a good compromise in terms of rate-complexity trade-off.

Overall, in comparison with other state-of-the-art encoders, all the proposed methods present higher compression ratios, reducing the average bit rates from 10 to 32%. The highest performance is achieved by DT-4D-MRP which to the best of the authors’ knowledge is the best lossless coding performance for LFs represented in the 4D spatial-angular domain achieved so far.

B. Future Work

A few research directions are left open by this paper and are intended to be addressed in the future. First and foremost, the authors aim to address the computational complexity of DT-4D-MRP, both by proper selecting the input parameters, as was done in [61] and also through a comprehensive analysis of the exact causes of this encoder computational complexity, aiming to propose directed solutions for this issue.

A previous work has shown that the bulk of the MRP bitstream is used to represent the prediction residuals [47]. Therefore, it should be worth it to investigate methods to improve the arithmetic encoding of these residuals. One idea that has not yet been explored is to forgo the use of a discretised set of probability models in favour of more modern methods, such as the adaptive learning of the probability models from previously decoded symbols.

REFERENCES

[1] G. Lippmann, “Epreuves reversibles donnant la sensation du relief,” J. Phys. Theor. Appl., vol. 7, no. 1, pp. 821–825, Jan. 1908.

[2] J. Wang, X. Xiao, H. Hua, and B. Javidi, “Augmented reality 3d displays with micro integral imaging,” Journal of Display Technology, vol. 11, no. 11, pp. 889–893, Nov. 2015.

[3] J. Arau, M. Kawakita, T. Yamashita, H. Sasaki, M. Miura, H. Hiura, M. Okui, and F. Okano, “Integral three-dimensional television with video system using pixel-offset method,” Opt. Express, vol. 21, no. 3, pp. 3474–3485, Feb. 2013.

[4] D. Shin, M. Cho, and B. Javidi, “Three-dimensional optical microscopy using axially distributed image sensing,” Opt. Lett., vol. 35, no. 21, pp. 3646–3648, Nov. 2010.

[5] P. M. Pereira, R. Fonseca-Pinto, R. P. Paiva, P. A. Assuncao, L. M. Tavora, L. A. Thomaz, and S. M. Faria, “Dermoscopic skin lesion image segmentation based on local binary pattern clustering: Comparative study,” Biomedical Signal Processing and Control, vol. 59, p. 101765, May 2020.

[6] ——, “Skin lesion classification enhancement using border-line features – the melanoma vs nevus problem,” Biomedical Signal Processing and Control, vol. 57, p. 101765, Mar. 2020.

[7] W. Zhou, E. Zhou, G. Liu, L. Lin, and A. Lumsdaine, “Unsupervised monoclonal depth estimation from light field image,” IEEE Trans. Image Process., vol. 29, pp. 1606–1617, Oct. 2020.

[8] K. Mishiba, “Fast depth estimation for light field cameras,” IEEE Trans. Image Process., vol. 29, pp. 4232–4242, Feb. 2020.

[9] R. Lourenc, P. A. A. Assuncao, L. M. N. Tavora, R. Fonseca-Pinto, and S. M. M. Faria, “Silhouette enhancement in light field disparity estimation using the structure tensor,” in IEEE International Conference on Image Processing (ICIP), Athens, Greece, Oct. 2018, pp. 2580–2584.

[10] H. Jung, H. Lee, and C. E. Rhee, “Flexibly connectable light field system for free view exploration,” IEEE Trans. Multimedia, vol. 22, no. 4, pp. 980–991, Apr. 2020.
[11] Y. Wang, J. Yang, Y. Guo, C. Xiao, and W. An, “Selective light field refocusing for camera arrays using bokeh rendering and superresolution,” IEEE Signal Process. Lett., vol. 26, no. 1, pp. 204–208, Jan. 2019.

[12] “JPG Pleno Call for Proposals on Light Field Coding,” Call for Proposals, International Organization for Standardization (ISO), Geneva, Switzerland, Jan. 2017.

[13] “Technical report of the joint ad hoc group for digital representations of light/sound fields for immersive media applications,” Technical Report, International Organization for Standardization (ISO), Geneva, Switzerland, Jun. 2016.

[17] J. M. Santos, P. A. A. Assunção, and S. M. M. Faria, “Exploration Studies Set #4 on JPEG Pleno Light Field Coding,” Doc. ISO/IEC JTC 1/SC29/WG1 N84025, International Organization for Standardization (ISO), Geneva, Switzerland, Mar. 2019.

[20] J. M. Santos, P. A. A. Assunção, L. A. da Silva Cruz, L. Tavora, R. Fonseca-Pinto, and S. M. M. Faria, “Lossless compression of light fields using multi-reference minimum rate predictors,” in Data Compression Conference (DCC), Snowbird, UT, USA, Mar. 2019, pp. 408–417.

[23] I. Matsuura, Y. Umez, N. Ozaki, J. Maeda, and S. Itoh, “A lossless coding scheme using adaptive predictors and arithmetic code optimized for each image,” Systems and Computers in Japan, vol. 38, no. 4, pp. 1–11, Mar. 2007.

[26] C. Zhou and S. K. Nayar, “Computational cameras: Convergence of fields using multi-reference minimum rate predictors,” in Data Compression Conference (DCC), Snowbird, UT, USA, Mar. 2019, pp. 408–417.

[30] I. Matsuda, H. Mori, J. Maeda, and S. Itoh, “A lossless coding scheme using adaptive predictors and arithmetic code optimized for each image,” Systems and Computers in Japan, vol. 38, no. 4, pp. 1–11, Mar. 2007.

[33] J. Matsuura, Y. Umez, N. Ozaki, J. Maeda, and S. Itoh, “A lossless coding scheme using adaptive predictors and arithmetic code optimized for each image,” Systems and Computers in Japan, vol. 38, no. 4, pp. 1–11, Mar. 2007.

[36] ——, “Residual-error prediction based on deep learning for lossless image compression,” Electronics Letters, vol. 54, no. 17, pp. 1032–1034, Aug. 2018.

[39] P. Helin, P. Astola, B. Rao, and I. Tabus, “Sparse modelling and predictive coding of subaperture images for lossless plenoptic image compression,” in 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video (3DTV-CON), Hamburg, Germany, Jul. 2016, pp. 1–4.

[42] ——, “Minimum description length sparse modeling and region merging for lossless plenoptic image compression,” IEEE J. Sel. Topics Signal Process., vol. 11, no. 7, pp. 1146–1161, Oct. 2017.

[45] I. Schiopu and A. Munteanu, “Macro-pixel prediction based on convolutional neural networks for lossless light-field compression,” in IEEE International Conference on Image Processing (ICIP), Athens, Greece, Oct. 2018, pp. 445–449.

[47] ——, “Deep-learning-based macro-pixel synthesis and lossless coding of light field images,” APSIPA Transactions on Signal and Information Processing, vol. 8, pp. 1–20, Jul. 2019.

[49] ——, “Macro-pixel prediction based on convolutional neural networks for lossless light-field compression,” in IEEE International Conference on Image Processing (ICIP), Athens, Greece, Oct. 2018, pp. 445–449.
[53] M. Rerabek and T. Ebrahimi, “New Light Field Image Dataset,” in International Conference on Quality of Multimedia Experience (QoMEX), Lisbon, Portugal, Jun. 2016.

[54] “Information technology - Plenoptic image coding system (JPEG Pleno) - Part 2: Light field coding,” Standard, International Organization for Standardization (ISO), Geneva, Switzerland, Apr. 2021.

[55] D. Flynn, D. Marpe, M. Naccari, T. Nguyen, C. Rosewarne, K. Sharman, J. Sole, and J. Xu, “Overview of the Range Extensions for the HEVC Standard: Tools, Profiles, and Performance,” IEEE Trans. Circuits Syst. Video Technol., vol. 26, no. 1, pp. 4–19, Sep. 2016.

[56] M. Briano and J. Serra-Sagristà, “CALIC, 3D CALIC and M-CALIC customizable software implementation,” https://prezi.com/j2ichhip4ls7/calic-implementation/, Jul. 2014, [Accessed 10 October 2021].

[57] M. J. Weinberger, G. Seroussi, and G. Sapiro, “The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS,” IEEE Trans. Image Process., vol. 9, no. 8, pp. 1309–1324, Aug. 2000.

[58] “Information technology - Lossless and near-lossless compression of continuous-tone still images,” Standard, International Organization for Standardization (ISO), Geneva, Switzerland, Jun. 1997.

[59] “Information technology - High efficiency coding and media delivery in heterogeneous environments - Part 2: High efficiency video coding,” Standard, International Organization for Standardization (ISO), Geneva, Switzerland, Aug. 2020.

[60] “Information technology - Coded representation of immersive media - Part 3: Versatile video coding,” Standard, International Organization for Standardization (ISO), Geneva, Switzerland, Aug. 2020.

[61] J. M. Santos, P. A. A. Assunção, L. A. da Silva Cruz, L. M. N. Távora, R. Fonseca-Pinto, and S. M. M. Faria, “Rate-complexity trade-off in minimum rate predictors light field lossless encoding,” in EURASIP European Light Field Imaging Workshop (ELFI), Borovets, Bulgaria, Jun. 2019, pp. 1–4.