Light Field Image Coding using High Order Prediction Training

Ricardo J. S. Monteiro, Paulo J. L. Nunes
Instituto de Telecomunicações, Lisbon, Portugal
ISCTE – Instituto Universitário de Lisboa (ISCTE-IUL),
Lisbon, Portugal
{ricardo.monteiro, paulo.nunes}@lx.it.pt

Sérgio M. M. Faria, Nuno M. M. Rodrigues
Instituto de Telecomunicações, Leiria, Portugal
ESTG, Instituto Politécnico de Leiria,
Leiria, Portugal
{sergio.faria,nuno.rodrigues}@co.it.pt

Abstract—**This paper proposes a new method for light field image coding relying on a high order prediction mode based on a training algorithm. The proposed approach is applied as an Intra prediction method based on a two-stage block-wise high order prediction model that supports geometric transformations up to eight degrees of freedom. Light field images comprise an array of micro-images that are related by complex perspective deformations that cannot be efficiently compensated by state-of-the-art image coding techniques, which are usually based on low order translational prediction models.**

The proposed prediction mode is able to exploit the non-local spatial redundancy introduced by light field image structure and a training algorithm is applied on different micro-images that are available in the reference region aiming at reducing the amount of signaling data sent to the receiver. The training direction that generates the most efficient geometric transformation for the current block is determined in the encoder side and signaled to the decoder using an index. The decoder is therefore able to repeat the high order prediction training to generate the desired geometric transformation. Experimental results show bitrate savings up to 12.57% and 50.03% relatively to a light field image coding solution based on low order prediction without training and HEVC, respectively.

Keywords—Light Field Image Coding, HEVC, High Order Prediction Training, Geometric Transformations

I. Introduction

Standard cameras are composed of two main elements: the lens and the camera sensor, which allows to capture light hitting the camera sensor on specific spatial coordinates. In single tier lenslet Light Field (LF) cameras, a third element is added: the microlens array (MLA). The MLA allows the LF camera to also capture the angular information of light hitting the sensor [1]. Each microlens creates a micro-image (MI) on the sensor containing both spatial and angular information about the light hitting the sensor. Thus, the captured LF image contains 3D information about the scene, instead of a single 2D perspective.

Due to the additional information that is captured, various a posteriori image processing tasks may be performed, such as, refocusing and changing the perspective after the picture has been taken [1]. This richer content capturing technology based on LF has applications in 3D television [2], image recognition, and medical imaging [3].

However, specific LF image and video coding algorithms are necessary to deal with the large amount of data generated by LF cameras. The growing interest in LF and other imaging technologies, such as point-cloud and 360-degree video, has led both JPEG and MPEG Committees to address coding and representation of these new imaging modalities. The new activities are known as JPEG Pleno [4] and MPEG-I [5].

Depending on the position of the camera sensor and MLA relative to the main lens, different types of LF lenslet images can be generated [6]. There are two main camera models for LF lenslet cameras, usually referred to unfocused (UNF) and focused (FOC). The difference between the two models is that the UNF model has the sensor one focal distance away from the MLA [7], and the FOC model has the MLA focused on the main lens image plane. Thus, different types of LF images are generated depending on the camera model. For example, if a UNF model is used, only angular information is captured, this means that each pixel within each MI corresponds to a different angle or viewpoint [7]. If a FOC model is being used, each individual MI is in focus, thus, in this case, the correspondent LF image is very similar to an array of very small cameras. This allows the FOC model to capture more spatial information in exchange for angular information, when compared to the UNF model [6].

Several authors have tried to exploit the non-local spatial redundancy that exists in LF images, i.e., redundancy between different MIs. In [8], [9] the discrete cosine transform (DCT) and the discrete wavelet transform (DWT) are used to exploit this type of redundancy. This is done by applying a 3D-DCT to a stack of MI [8] or a 3D-DWT [9] to a stack of sub-aperture images (SAIs). Each SAI represents a rendered image, from a different perspective, extracted from the LF. By taking advantage of this alternative way to represent the LF, the SAIs can be re-organized into a pseudo-video sequence (PVS), which can then be compressed using a standard video encoder. The non-local redundancy of the LF image is exploited by inter prediction tools available in most video encoders. Several authors have proposed different scanning strategies to generate the PVS [10]–[12]. Raster and

The authors acknowledge the support of Fundação para a Ciência e Tecnologia, under the projects UID/EEA/50008/2013.

ISBN 978-90-827970-1-5 © EURASIP 2018

spiral scans were tested with H.264 [10] and HEVC [11]. In both cases it can be concluded that the most efficient scanning strategy is spiral. Recently, in [12], a new PVS scheme was proposed where the SAI are organized into layers, starting from the central SAI and moving on to the outer SAIs. The further away the layer is to the central SAI the higher the quantization parameter (QP) is for each layer.

Other authors proposed to add new prediction tools to existent image codecs, allowing the codec to exploit non-local redundancy [13]–[15]. In [13] a self-similarity (SS) compensated prediction is proposed that takes advantage of the flexible partition patterns used by HEVC. The authors in [14] extended this approach by developing a multi-hypothesis coding method using up to two hypotheses for prediction in spatial and time domain. In [15] a non-local spatial prediction method has been investigated that uses locally linear embedding combined with template matching. These approaches are able to vastly outperform HEVC for LF images.

The approaches based on SS can be considered low order prediction (LOP) approaches because they are limited to two translational degrees of freedom (DoF). This limitation reduces the prediction method ability to describe the changes in perspective between adjacent MIs. These changes in perspective require a geometric transformation (GT) with up to 8 DoF. In order to mitigate this limitation, the authors have proposed to integrate in HEVC a high order prediction (HOP) approach in [16] to exploit the non-local spatial redundancy. The proposed approach is able to not only outperform HEVC, but also, solutions based on the SS approach [13]. Although the added degrees of freedom (DoF) improve the coding efficiency when compared to SS, the amount of additional overhead necessary to describe the HOP is high when compared to a LOP approach, i.e., four additional vectors are transmitted, per block, to the decoder.

In this paper, a new HOP approach is proposed, that is able to estimate an efficient GT using a HOP training stage applied to a causal area of the LF image. The proposed approach is applied as an Intra prediction method based on a two-stage block-wise HOP model. In the first stage, a LOP approach is used to generate an approximated prediction block. In the second stage, HOP training is applied using several training directions in the causal area of the LF image. Each training direction corresponds to the location of adjacent MIs that are already encoded, e.g., the upper, left and upper-left MIs. Since adjacent MIs are typically very similar, it is expected that the GT that is generated from the training step, will also produce an efficient prediction block. The most efficient training direction is transmitted to the decoder as an index. Since the training is performed in the causal area of the LF image, the GT for the second stage of the proposed HOP approach can be also calculated on the decoder side. Therefore, no overhead is necessary to describe the second stage of the HOP approach, but the most efficient training direction index. By taking advantage of the extra DoF in HOP models and the reduced overhead, the proposed approach is able to outperform state-of-the-art techniques based on LOP models and in some cases the proposed approach in [16].

The remainder of this paper is organized as follows: Section II describes the HOP training algorithm; Section III reviews the HOP model; Section IV presents the experimental results; and, finally, Section V concludes the paper.

II. High Order Prediction Training

In this section the proposed HOP training algorithm is described. This algorithm is integrated in HEVC as an intra prediction mode in conjunction with the Planar, DC and the 33 Directional modes. The proposed HOP training algorithm can be described through Algorithm 1.

Algorithm 1 High Order Prediction Training

```
Input current block; reference region
Output T vector; HOP training index
Apply LOP model to generate T vector
Generate prediction block $B_{LOP}$ using T vector
Estimate cost, $J_{LOP}$, for transmitting the LOP model information, i.e., vector T and a null HOP training index
Generate list of training directions and find $B_t$ blocks in $n$ adjacent MIs
for each training direction; do
    Apply HOP model (Algorithm 2) using $B_{t_n}$ and the reference region to generate a GT candidate
    Apply GT candidate to $B_{LOP}$ in order to generate the prediction block, $B_{HOP_n}$
    Estimate cost, $J_{HOP_n}$, for transmitting the HOP information necessary to generate $B_{HOP_n}$, i.e., vector T and HOP training index $(n+1)$
end for
Encode HOP information that corresponds to the lowest cost among $J_{LOP}$ and $J_{HOP_n}$ candidates
```

In the first stage, a LOP search is used between the current block and the reference region. A full search algorithm is used, as proposed in [13], and the output is a translational vector $\boldsymbol{T}$ (2 DoF). This option is available in case the HOP training is not able to find a “good” GT candidate. The efficiency of the HOP training tends to be higher when the redundancy between the current block and at least one of the B_t blocks, available in each training direction, is high.

The proposed algorithm can be applied to any number of training directions. The goal is to find a training direction that minimizes the rate-distortion (RD) cost of the generated prediction block, $B_{HO\ \ n}$. Since the HOP training index is transmitted to the decoder, only the training that provides the best result is repeated on the decoder side. However, the number of bits necessary to transmit the HOP training index increases with the number of training directions. The training directions are selected based on the proximity to the current block. For example, for three training directions, i.e., $n = 3$, using a square-based MLA, the blocks B_t are located in: $B_{t_0} = (-m, 0)$; $B_{t_1} = (0, -m)$ and $B_{t_2} = (-m, -m)$. Where m is the size of the MI in pixels. These locations correspond to estimated locations of the block B_{t_n} in the left, upper and upper-left MIs.

For each of the defined training directions, the HOP model is applied (Algorithm 2), using as an input block B_{t_n} and the reference region. Algorithm 2 is explained in detail in Section III. The output of the HOP model estimation is a GT candidate per training direction. Each GT candidate is therefore applied to the prediction block, B_{LOP}, generating n different B_{HOP_n} prediction blocks. To determine which training direction is the one that generates the most efficient prediction block, an RD cost value is calculated for each of the n training directions.

The cost is calculated using the Lagrangian cost $J = D + \lambda R$, where D is the distortion between each of the generated prediction blocks (B_{LOP} and B_{HOP_n}) and the current block, respectively. The distortion is calculated using the sum of absolute differences in the Hadamard domain (SADHAD) as it was used in [16]. The λ value is the Lagrange multiplier, computed as in HM version 15.0 for Intra coded frames. The estimated rate necessary to encode the HOP information, R, is the estimated number of bits necessary to encode the vector $\boldsymbol{T}$ and the HOP training index. The rate associated with vector $\boldsymbol{T}$ is estimated like the motion vector rate in HM version 15.0, while the rate estimate for the HOP training index is given by $\log_2(n + 1)$, where n is the number of training directions.

Once all the costs J_{LOP} and J_{HOP_n} associated to each possible candidate are generated, the lowest one corresponds to the most RD efficient. The most efficient option is then compared to the cost of the other intra prediction modes, i.e., DC, Planar, and the 33 Directional modes, and the mode with the lowest RD cost is selected and encoded. Vector $\boldsymbol{T}$ and the HOP training index are transmitted to the decoder using the HM approach for motion vectors and motion vector prediction index [16], and encoded using the context adaptive binary arithmetic coding (CABAC) entropy coding method.

III. High Order Prediction Model

This section describes the HOP model that is applied in this paper. The HOP model aims to find the best GT that matches two quadrilaterals. In [16] this model was used to find a GT that matches the current block being encoded with a corresponding block in the reference region, the causal area of pixels already encoded. However, since the current block is not yet available at the decoder side, the GT information needs to be transmitted. In this paper, the HOP model is used to find a GT that matches two quadrilaterals, both inside the reference region, in order to be integrated in the training algorithm. Since both blocks are part of the reference region, the decoder is able to repeat the same training algorithm and generate the same GT. The two quadrilaterals are the block B_t corresponding to each training direction, as mentioned in the previous section, and an arbitrary quadrilateral (Q_t) inside a specified search window. Algorithm 2 describes the HOP model estimation.

Algorithm 2 High Order Prediction Model Estimation

```
Input $B_t$; reference region
Output GT candidate
1. Generate a list of correspondence points between
   the corners of $B_t$ and $Q_t$
2. for each correspondence point; do
3.     Calculate the GT parameters between $B_t$ and $Q_t$
4.     Apply inverse GT mapping to $Q_t$ to generate a
       prediction block $B_{HOP}$ with the same shape as $B_t$
5.     Calculate distortion between $B_{HOP}$ and $B_t$
6. end for
7. Select the GT that achieves the lowest distortion
```

The first step of the HOP model estimation allows the creation of a list of correspondence points between the corners of B_t and Q_t to be tested. Ultimately, every combination of points can be tested within the search window, however if the window size is, e.g., $W = 128$ pixels, the number of combinations would be $(2W^2)^4$, i.e., more than 1.15×10^{18}, which is impractical. To reduce the number of combinations, a two-stage process is used where in the first stage a LOP search is applied and in the second stage a HOP search is applied centered in the best result of the first stage.

Since the LOP search has been already performed, i.e., in Algorithm 1 (see Section II), vector $\boldsymbol{T}$ is therefore directly applied to the spatial position of block B_t. The second stage uses a HOP search algorithm based on a 2D logarithmic fast search algorithm applied to the four corners of Q_t [16].

The vectors that connect the corners of B_t and Q_t are defined by:

$$\begin{cases} \vec{v}_{HOP_0} = (u_0, v_0) + \boldsymbol{T} \\ \vec{v}_{HOP_1} = (u_1 - (B_x - 1), v_1) + \boldsymbol{T} \\ \vec{v}_{HO\ \ 2} = (u_2 - (B_x - 1), v_2 - (B_y - 1)) + \boldsymbol{T} \\ \vec{v}_{HOP_3} = (u_3, v_3 - (B_y - 1)) + \boldsymbol{T} \end{cases} \tag{1}$$

where B_x and B_y are the width and height of B_t, respectively. The values u_n and v_n are the relative corner displacements of block Q_t. Every combination of points of correspondence are defined by the system of equations in (1).

To generate the prediction block, B_{HOP}, it is first necessary to calculate the GT between B_t and Q_t as described by step 3 in Algorithm 2. This can be done using a Projective GT, which can be defined by a 3×3 matrix $\boldsymbol{H}$ verifying (2):

$$[x, y, 1] = [uh, vh, h]\boldsymbol{H}\ . \tag{2}$$

The x and y values correspond to the pixel positions of B_{HOP}, and u and v correspond to the pixel position of B_t. The Projective matrix $\boldsymbol{H}$ can be decomposed into three different submatrices, $\boldsymbol{L_p}$, $\boldsymbol{T_p}$ and $\boldsymbol{P_p}$:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{L_p} & \boldsymbol{P_p} \\ \boldsymbol{T_p} & 1 \end{bmatrix}$$
$$\boldsymbol{L_p} = \begin{bmatrix} l_{00} & l_{01} \\ l_{10} & l_{11} \end{bmatrix}, \boldsymbol{T_p} = \begin{bmatrix} t_x & t_y \end{bmatrix}, \boldsymbol{P_p^T} = \begin{bmatrix} p_x & p_y \end{bmatrix} \tag{3}$$

Each submatrix can be calculated as:

$$\boldsymbol{P_p} = \begin{bmatrix} \dfrac{1 \begin{vmatrix} \Delta u_3 & \Delta u_2 \\ \Delta v_3 & \Delta v_2 \end{vmatrix}}{B_x - 1 \begin{vmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{vmatrix}} \\ \dfrac{1 \begin{vmatrix} \Delta u_1 & \Delta u_3 \\ \Delta v_1 & \Delta v_3 \end{vmatrix}}{B_y - 1 \begin{vmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{vmatrix}} \end{bmatrix},$$
$$\boldsymbol{L_p} = \begin{bmatrix} \frac{u_1 - u_0}{B_x - 1} + p_x u_1 & \frac{u_3 - u_0}{B_y - 1} + p_y u_3 \\ \frac{v_1 - v_0}{B_x - 1} + p_x v_1 & \frac{v_3 - v_0}{B_y - 1} + p_y v_3 \end{bmatrix} \text{ and} \tag{4}$$
$$\boldsymbol{T_p} = \boldsymbol{T} + \begin{bmatrix} u_0 & v_0 \end{bmatrix},$$

where:

$$\begin{cases} \Delta u_1 = u_1 - u_2 \\ \Delta u_2 = u_3 - u_2 \\ \Delta u_3 = u_0 - u_1 + u_2 - u_3 \end{cases}$$
$$\begin{cases} \Delta v_1 = v_1 - v_2 \\ \Delta v_2 = v_3 - v_2 \\ \Delta v_3 = v_0 - v_1 + v_2 - v_3 \end{cases}. \tag{5}$$

Since the GT parameters are now available, the prediction block, B_{HOP}, can be generated by applying an inverse mapping to Q_t, as described in step 4 of Algorithm 2. Inverse mapping is applied because the generated prediction block, B_{HOP}, should have the same shape and size as B_t, so it can be compared using a distortion metric [16]. For each pixel position of B_{HOP}, i.e., u and v, it is necessary to calculate the spatial position inside the reference region where that pixel is located, i.e., x and y. This can be calculated by turning (2) into the system of equations (6):

$$\begin{cases} x = \dfrac{l_{00}u + l_{10}v + t_x}{p_x u + p_y v + 1} \\ y = \dfrac{l_{01}u + l_{11}v + t_y}{p_x u + p_y v + 1} \end{cases} \tag{6}$$

When the values x and y are not integers, bilinear interpolation is used to calculate the pixel value in that position. After applying the system of equations (6) to all the pixels inside B_{HOP}, i.e., $u \in [0, B_x - 1]$ and $v \in [0, B_y - 1]$, the prediction block can be generated and compared with B_t by calculating the distortion, as described in step 5 of Algorithm 2. As in Algorithm 1, the distortion metric used is the SADHAD.

Steps 3 to 5 are repeated for all the correspondence point generated by the 2D logarithmic fast search algorithm. Finally, in step 7, the GT that generates the lowest distortion is selected as a GT candidate.

IV. Experimental Results

In this section the experimental results of the proposed coding solution for LF image coding, that implements the proposed HOP training is evaluated and compared against state-of-the-art solutions based on LOP and HOP approaches.

To evaluate the performance of the proposed HOP training solution, 6 LF images captured using a FOC model and a square-based MLA are used. The selected images for testing have different resolutions and MI resolutions. *Plane and Toy* images, frame 0 (PT0) and 150 (PT150), have a resolution of 1920×1088 (MIs 28×28); *Demichelis Spark* (DS) and *Demichelis Cut* (DC) images, frame 0, have a resolution of 2880×1620 (MIs 38×38); *Laura* and *Seagull* have a resolution of 7240×5432 (MIs 75×75). Additionally, a subset of the EPFL dataset is also used for testing which is composed of 12 LF images. These images were captured using a Lytro Illum, therefore the camera model is UNF and uses a hexagon-based MLA. In this case the images have a resolution of 7728×5368 pixels (MIs 15×15).

All the images were encoded and decoded using a modified implementation of HM version 15.0 that includes the additional proposed prediction mode, i.e., the HOP training. This implementation is referred to as HEVC-HOP-nT. As mentioned above the proposed HOP training can be applied to any number of training directions, therefore the n represents the number of training directions available. The authors compare HEVC-HOP-nT with HEVC, i.e., where only the standard Intra modes are available. Additionally, the work in [13] and [16] referred to as HEVC-SS and HEVC-HOP, respectively, are also used for comparison. HEVC-SS represents a solution based on LOP and HEVC-HOP represents a solution based on HOP. All images are encoded using the common HM test conditions, using QP values of 22, 27, 32 and 37 and a causal window of 128 pixels, for HEVC-SS, HEVC-HOP and HEVC-HOP-nT.

To evaluate the performance of all the codecs the Bjøntegaard Delta Metric is used. The experimental results for the above-mentioned test images are shown in Table I.

When comparing HEVC with HEVC-SS, which is limited to only 2 DoF, it is possible to see that HEVC-SS is able to outperform HEVC with average bitrate savings of 27.91% and 23.11% for LF images using FOC and UNF camera models, respectively. However, when comparing HEVC-HOP, which supports up to 8 DoF, with HEVC-SS, additional average bitrate savings of 7.47% and 5.06% are achieved for LF images using FOC and UNF camera models, respectively.

When comparing the results for the proposed HEVC-HOP-nT it is possible to see that the bitrate savings increase for every image when the number of training directions is increased. For seven training directions, i.e., HEVC-HOP-7T,

TABLE I
BD-PSNR-Y AND BD-RATE RESULTS COMPARING HEVC, HEVC-SS, HEVC-HOP AND HEVC-HOP-T, USING ONE, THREE AND SEVEN TRAINING DIRECTIONS

Image	HEVC-SS vs HEVC		HEVC-HOP vs HEVC-SS		HEVC-HOP-1T vs HEVC-SS		HEVC-HOP-3T vs HEVC-SS		HEVC-HOP-7T vs HEVC-SS	
	BD-PSNR-Y	BD-RATE	BD- PSNR-Y	BD-RATE	BD- PSNR-Y	BD-RATE	BD- PSNR-Y	BD-RATE	BD- PSNR-Y	BD-RATE
PT0	0.90 dB	-14.64 %	**0.27 dB**	**-4.66 %**	0.07 dB	-1.20 %	0.12 dB	-2.18 %	0.17 dB	-3.06 %
PT150	1.44 dB	-19.02 %	**0.75 dB**	**-11.05 %**	0.28 dB	-4.12 %	0.47 dB	-6.98 %	0.56 dB	-8.30 %
DS	1.09 dB	-31.43 %	0.26 dB	-8.39 %	0.18 dB	-5.71 %	0.24 dB	-7.50 %	**0.29 dB**	**-9.14 %**
DC	1.05 dB	-29.25 %	**0.30 dB**	**-9.14 %**	0.17 dB	-5.41 %	0.24 dB	-7.43 %	0.30 dB	-9.09 %
Laura	2.26 dB	-30.35 %	0.27 dB	-4.78 %	0.15 dB	-2.65 %	0.34 dB	-6.11 %	**0.40 dB**	**-7.09 %**
Seagull	2.81 dB	-42.78 %	0.31 dB	-6.82 %	0.23 dB	-5.09 %	0.51 dB	-11.08 %	**0.59 dB**	**-12.57 %**
AVG. FOC	1.59 dB	-27.91 %	0.36 dB	-7.47 %	0.18 dB	-4.03 %	0.32 dB	-6.88 %	**0.39 dB**	**-8.21 %**
AVG. EPFL	0.83 dB	-23.11 %	**0.14 dB**	**-5.06 %**	0.02 dB	-0.65 %	0.04 dB	-1.34 %	0.06 dB	-2.14 %

the achieved average bitrate savings, for the LF images captured by a camera using a FOC model when compared to HEVC-SS, is 8.21% (up to 12.57%). Additionally, when compared to HEVC the average bitrate savings is 33.55% (up to 50.03%). In this case, HEVC-HOP-7T is able to outperform HEVC-HOP. However, when encoding images from the EPFL dataset using HEVC-HOP, the average bitrate savings relatively to HEVC-SS is 5.06%, where when the proposed HEVC-HOP-7T is used, average bitrate savings of only 2.14% are achieved. Concluding that for LF images captured with a UNF camera model, HEVC-HOP is more efficient than the proposed HEVC-HOP-7T. The number of training directions can be further increased, however the bitrate savings gains for a higher number of training directions is residual and the encoder computational complexity is vastly increased.

Since the proposed approach, i.e., HEVC-HOP-nT, can be more efficient than HEVC-HOP for LF images captured with a FOC camera model and HEVC-HOP is more efficient for LF images captured with an UNF camera model, a hybrid codec could be implemented that is able to use both prediction modes. Such hybrid codec will be investigated as future work.

V. CONCLUSIONS

In this paper a HOP mode based on a training algorithm was proposed. The proposed approach is applied as an Intra prediction method based on a two-stage block-wise HOP model that supports GTs up to 8 DoF. The proposed approach is able to vastly outperform HEVC and a LOP coding solutions. Experimental results using LF images captured by FOC camera models, show average bitrate savings of 8.21% and 33.55% relatively to the LOP coding and HEVC, respectively. The proposed HOP mode based on a training algorithm is able to outperform the HOP coding solution [16] with the same support for 8 DoF for this type of LF images. However, when encoding LF images captured by UNF camera models, although the proposed solution is able to outperform both HEVC and the LOP coding solution, with average bitrate savings of 24.28% and 2.14%, respectively, it is not able to outperform the HOP coding solution [16]. Nonetheless, bitrate savings increase consistently with the number of training directions. Additionally, the authors concluded experimentally that for more than seven training directions the bitrate saving gains start decreasing.

Future work will include the study of a hybrid codec that implements both HOP coding solutions.

REFERENCES

[1] T. Georgiev and A. Lumsdaine, "Rich Image Capture with Plenoptic Cameras," in *IEEE Inter. Conf. on Computational Photography*, Cluj-Napoca, Romania, 2010, pp. 1–8.

[2] J. Arai, "Integral three-dimensional television (FTV Seminar)," Sapporo, Japan, ISO/IEC JTC1/SC29/WG11 MPEG2014/N14552, Jul. 2014.

[3] X. Xiao, B. Javidi, M. Martinez-Corral, and A. Stern, "Advances in three-dimensional integral imaging: sensing, display, and applications," *Appl Opt*, vol. 52, no. 4, pp. 546–560, Feb. 2013.

[4] "JPEG PLENO Abstract and Executive Summary," Sydney, ISO/IEC JTC 1/SC 29/WG1 N6922, Feb. 2015.

[5] "MPEG-I Technical Report on Immersive Media," Torino, Italy, ISO/IEC JTC1/SC29/WG11 N17069, Jul. 2017.

[6] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *IEEE Inter. Conf. on Computational Photography*, 2009, pp. 1–8.

[7] C. Hahne, A. Aggoun, S. Haxha, V. Velisavljevic, and J. C. J. Fernández, "Light field geometry of a standard plenoptic camera," *Opt Express*, vol. 22, no. 22, pp. 26659–26673, Nov. 2014.

[8] A. Aggoun, "A 3D DCT Compression Algorithm For Omnidirectional Integral Images," in *IEEE Inter. Conf. on Acoustics, Speech and Signal Processing*, 2006, vol. 2, pp. II–II.

[9] A. Aggoun, "Compression of 3D Integral Images Using 3D Wavelet Transform," *J. Disp. Technol.*, vol. 7, no. 11, pp. 586–592, Nov. 2011.

[10] F. Dai, J. Zhang, Y. Ma, and Y. Zhang, "Lenselet image compression scheme based on subaperture images streaming," in *IEEE Inter. Conf. on Image Processing*, 2015, pp. 4733–4737.

[11] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of Lytro-Illum light fields," in *Inter. Conf. on Image Processing Theory, Tools and Applications*, 2015, pp. 494–497.

[12] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *IEEE Inter. Conf. on Multimedia Expo Workshops*, 2016, pp. 1–4.

[13] C. Conti, L. D. Soares, and P. Nunes, "HEVC-based 3D holoscopic video coding using self-similarity compensated prediction," *Signal Process. Image Commun.*, vol. 42, pp. 59–78, Mar. 2016.

[14] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Coding of Focused Plenoptic Contents by Displacement Intra Prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1308–1319, Jul. 2016.

[15] L. F. R. Lucas *et al.*, "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC," in *European Signal Processing Conf.*, Lisbon, Portugal, 2014, pp. 11–15.

[16] R. J. Monteiro, P. Nunes, N. Rodrigues, and S. M. M. de Faria, "Light Field Image Coding using High Order Intra Block Prediction," *IEEE J. Sel. Top. Signal Process.*, vol. 11, no. 7, pp. 1120-1131, Oct. 2017.