Light Field Image Coding Using High-Order Intrablock Prediction

Ricardo J. S. Monteiro, *Member, IEEE*, Paulo J. L. Nunes, *Member, IEEE*, Nuno M. M. Rodrigues, *Member, IEEE*, and Sérgio M. M. Faria, *Senior Member, IEEE*

***Abstract*—This paper proposes a two-stage high-order intrablock prediction method for light field image coding. This method exploits the spatial redundancy in lenslet light field images by predicting each image block, through a geometric transformation applied to a region of the causal encoded area. Light field images comprise an array of microimages that are related by complex geometric transformations that cannot be efficiently compensated by state-of-the-art image coding techniques, which are usually based on low-order translational prediction models. The two-stage nature of the proposed method allows us to choose the order of the prediction model most suitable for each block, ranging from pure translations to projective or bilinear transformations, optimized according to an appropriate rate-distortion criterion. The proposed higher order intrablock prediction approach was integrated into a high efficiency video coding (HEVC) codec and evaluated for both unfocused and focused light field camera models, using different resolutions and microlens arrays. Experimental results show consistent bitrate savings, which can go up to 12.62%, when compared to a lower order intrablock prediction solution and 49.82% when compared to HEVC still picture coding.**

***Index Terms*—Geometric transformations, HEVC, high order intra block prediction, light field image coding.**

I. Introduction

LIGHT Field (LF) imaging technology available in lenslet LF cameras allows to jointly capture radiance data and angular information from the light rays hitting the camera's sensor, by means of multiplexing the LF data in a 2D conventional sensor. This is achieved through an array of microlenses, placed between the main lens and the camera sensor. Each microlens creates a micro-image (MI) on the sensor, which is the microlens scene perspective being captured through the main lens. Therefore, a lenslet light field image tends to be like the output of an array of very small cameras.

The additional knowledge of the scene angular information allows to perform various a posteriori image processing tasks, not straightforwardly possible with traditional cameras. Refocusing and change of perspective after the picture has been taken are the most common examples [1]. These functionalities, derived from the ability to capture the "whole observable" (LF) scene [1], may be advantageous for several applications, like 3D Television [2], since by rendering several views from different perspectives, 2D, 3D and multiview signals can be created; image recognition and medical imaging [3].

Depending on the position of the camera sensor and the microlens array relatively to the main lens, different samplings of the light field can be performed, which define essentially the lenslet LF camera model [4]. Two main types of lenslet LF camera models exist, the unfocused [5] and the focused model [4]. In the classic unfocused camera model case, the sensor is one focal distance away from the microlens array. Thus, the microlens array is focused at infinity, i.e., the light rays that reach the microlens array are parallel [5]. Consequently, the microlens array is completely defocused from the main lens image plane. Therefore, each microlens only captures angular information, meaning that each pixel, within the MI, corresponds to a different angle, or viewpoint [5]. In the focused lenslet LF camera model, the sensor is away from the microlens array focal distance and the microlens array is focused on the main lens image plane, allowing for each microlens to generate a focused MI. This feature allows a higher spatial resolution for rendering, since more than one pixel can be extracted from each MI in the rendering process [4]. These models have been the base for the deployment of this technology, allowing an increasing number of applications and users.

The growing interest in LF technology led the JPEG Committee to launch a new activity, known as JPEG Pleno, to address coding and representation of content generated by emerging imaging technologies such as LF, point-cloud and holographic technologies [6].

The large amount of data required to adequately represent a LF scene, when compared to the case of typical 2D pictures, calls for efficient techniques for both transmission and storage of this type of content. In this context, several authors proposed specific LF coding techniques, which can be applied directly to the lenslet LF images, in order to exploit the MIs redundancy. Alternatively, other techniques are applied to a different representation of the same LF, which comprises the view point

Manuscript received December 29, 2016; revised April 15, 2017; accepted June 7, 2017. Date of publication June 28, 2017; date of current version October 23, 2017. This work was supported by the Fundação para a Ciência e Tecnologia under Project UID/EEA/50008/2013. The guest editor coordinating the review of this paper and approving it for publication was Prof. Yebin Liu. *(Corresponding author: Ricardo J. S. Monteiro.)*

R. J. S. Monteiro and P. J. L. Nunes are with the Instituto de Telecomunicações, 1049-001 Lisbon, Portugal, and the University Institute of Lisbon, 1649-026 Lisbon, Portugal (e-mail: ricardo.monteiro@lx.it.pt; paulo.nunes@lx.it.pt).

N. M. M. Rodrigues and S. M. M. Faria are with the Instituto de Telecomunicações, 2411-901 Leiria, Portugal, and the Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, 2411-901 Leiria, Portugal (e-mail: nuno.rodrigues@co.it.pt; sergio.faria@co.it.pt).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2017.2721358

images, also known as sub-aperture images (SAIs). The SAIs are generated by extracting at least one pixel, in a fixed position, from each MI and organizing them into a matrix. Each SAI represents a rendered image, from a different perspective, extracted from the LF image.

State-of-the-art LF coding schemes rely on block matching techniques to exploit the inherent spatial redundancy in lenslet LF images. However, these low order prediction (LOP) models use only two degrees of freedom (DoF), as only translations are used to describe the inherent LF image spatial redundancy.

Due to the small baseline between MIs in lenslet LF images, the different MIs can be approximately related by changes in perspective, which require eight DoF to be described. To the best of authors' knowledge, this characteristic has not been exploited by previous LF coding approaches described in the literature. This additional matching accuracy is important to develop a coding method able to cope with important features of the LF content, such as:

1) *The LF camera model,* i.e., both focused and unfocused models should be handled;
2) *The type of microlens array structure,* e.g., rectangular or hexagonal microlens layouts, creating rectangular, hexagonal or circular MIs;
3) *The MI size*, i.e., a parameter that depends on the camera, and has a strong influence on the number of possible rendered points of view and their spatial resolution.

High order prediction (HOP) models, e.g., using geometric transformations with more DoF, have been studied during the last two decades in traditional 2D and 3D image coding scenarios. Several geometric models, like translation, rotation, scale, shear and perspective changes have been used to improve the coding efficiency, by exploiting spatial [7], temporal [8]–[13] and inter-view [14]–[17] redundancy. In most proposals, these models have been applied image-wise (instead of block-wise), due to two main reasons: (i) high computational complexity in block-wise model parameter estimation, and (ii) significant additional bit rate required for parameter transmission. Despite these drawbacks, this paper demonstrates that block-wise HOP models can increase block matching accuracy and, thus, coding efficiency for lenslet LF images.

The method proposed in this paper for encoding lenslet LF images relies on a two-stage block-wise HOP model, where each image block is intra predicted from a reference in the causal area of the image, i.e., containing pixels that were already encoded. Since this approach is applied block-wise, it is possible to optimize the HOP model (number of DoF) for each block to be encoded. Taking advantage of the extra DoF available in HOP models, it is possible to outperform state-of-the-art coding techniques based on LOP models.

The remainder of this paper is organized as follows: Section II presents a review of several relevant state-of-the-art solutions, regarding LF image coding; Section III describes the geometric transformations used in the proposed prediction method; Section IV presents the proposed HOP model; Section V presents the test conditions and experimental results; and, finally, Section VI concludes the paper.

II. Related Work on Light Field Image Coding

Several schemes to encode lenslet LF images are described in the literature, aiming to exploit the intra-LF image redundancy. These schemes rely on different LF image representations and coding techniques, which may be categorized according to the fundamental adopted approach as: transform-based coding, pseudo-video sequence coding, disparity-based coding and non-local spatial prediction coding.

A. *Transform-Based Coding*

Some LF coding schemes rely, essentially, on the use of a transform, mainly the discrete cosine transform (DCT) [18], [19] or the discrete wavelet transform (DWT) [20]. In [18], a 3D-DCT is applied to a stack of MIs, to exploit the existing spatial redundancy within a MI, as well as the redundancy between adjacent MIs. In [20], a LF image is decomposed into SAIs, and a 3D-DWT is applied to a stack of these SAIs. The lower frequency bands are transformed using a two-dimensional discrete wavelet transform (2D-DWT), while the remaining higher frequency coefficients are simply quantized and arithmetic encoded. These coding schemes are reportedly more efficient than JPEG, but not as efficient as high efficiency video coding (HEVC) still picture coding.

B. *Pseudo-Video Sequence Coding*

This type of LF coding schemes represent the LF image as a set of MIs or SAIs, and re-organize them into a low resolution pseudo-video sequence (PVS), which is then compressed using a standard video encoder. Various scanning strategies to order the PVS are considered to better exploit the redundancy between MIs or SAIs. Dai *et al*. [21] propose to scan the SAIs using either a raster or a spiral scan and then encode the generated video sequence with H.264/AVC. Vieira *et al*. [22] used similar scanning strategy combined with several prediction structures supported by HEVC. In both cases it is possible to conclude that the spiral scan is more efficient than the raster scan. More recently, in the ICME light field compression challenge [23], Liu *et al*. [24] used a PVS scheme to organize the SAIs into layers, depending on the proximity to the central view, starting with the central SAI and moving on to the outer views. The more distant the SAI is from the center, the higher the value of the used quantization parameter (QP) should be. This scheme was implemented using both HEVC test model (HM) and JEM [25] software. Because the rate allocation is not uniform along the LF image, this method is prone to reconstruct views with different objective qualities.

C. *Disparity-Based Coding*

In this type of LF coding schemes the LF image is considered as a set of views captured by different cameras (either in the form of MIs or SAIs), which may be encoded exploiting inter-view disparity. In [26], the authors propose a coding method that uses some SAIs to calculate a set of disparity maps prior to coding, which are then used to predict the remaining SAIs. The authors concluded that this approach is suitable to encode synthetic images, where disparity compensation alone can be enough to predict a SAI. A compression scheme that incorporates disparity compensation into 4D wavelet coding using disparity compensated lifting is proposed in [27]. The disparity information derived from an approximated model of the scene is applied to modify the update and prediction filters of the lifting procedure. In [28], the authors propose a scalable (two-layer) LF coding approach for the focused LF camera model, using a LF representation that consists of a sparse set of MIs and associated disparity maps. Based on the sparse set of MIs and the associated disparity maps (first layer), a reference prediction LF image is obtained through a reconstruction method that relies on disparity-based interpolation and inpainting. This reconstructed LF image is then used to encode the original LF image (second layer), by encoding the prediction residue. This approach was later extended [29] with a third layer of scalability and the use of lossy encoded disparity maps, in contrast with the lossless transmission of the disparity maps, used in the first approach. Both versions of the work are able to outperform HEVC still picture coding.

D. Non-Local Spatial Prediction Coding

Several methods to exploit the non-local spatial redundancy were proposed as additional coding tools for existing video coding standards, like HEVC. In [30], a self-similarity compensated prediction is proposed to take advantage of the flexible partition patterns used by this video codec. In [31] this method was extended with a bi-directional mode to increase its coding efficiency. Additionally, in [32], an alternative non-local spatial prediction method has been investigated, relying on a prediction mode based on locally linear embedding integrated in HEVC. Differently from the other schemes, that exploit non-local spatial redundancy, this method distributes the computational complexity between the encoder and the decoder, i.e., the locally linear embedding procedure must be replicated in the decoder. In [33] the authors developed a multi-hypothesis coding method specifically for focused LF image and video. This method uses up to two hypotheses for prediction in both spatial and time domains, which outperforms single-hypothesis based prediction. For the unfocused camera model, the authors concluded that the rate-distortion efficiency is still much higher, compared to JPEG or HEVC, however the gains relatively to HEVC are smaller in this case when compared to the focused model [34].

The main advantage of this category is that, in most approaches, the lenslet LF images are encoded without the need of any pre-processing steps or any prior knowledge about the capturing device, e.g., the LF camera model, the microlens array structure and the MI size.

III. Geometric Transformations for High Order Prediction

In most state-of-the-art encoders, prediction between blocks of pixels is performed using very simple transformations, like translations. However, a lenslet LF image is comprised of MIs that are related by more complex transformations, resulting from the fact that each MI represents the scene being captured from slightly different perspectives. In such cases, it is advantageous

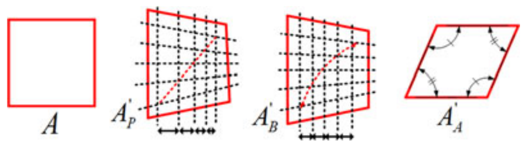

Fig. 1. Examples of possible GTs applied to block A: Projective (A'_P), bilinear (A'_B) and affine (A'_A).

to use geometric transformations that better exploit the features of the LF image and its MIs.

A geometric transformation (GT) is able to map perspective changes from one view (generically associated to a quadrilateral) into another view, requiring up to eight DoF.

Considering two different blocks, A and A', each one with its own coordinate system, (u, v) and (x, y), respectively, it is possible to define a generic relationship:

$$(x, y) = (X(u, v), Y(u, v)), \tag{1}$$

where X and Y are mapping functions for each coordinate. These functions create a point to point correspondence between images. Depending on the number of DoF used by the mapping functions in (1), different number of independent point to point correspondences are possible. To describe these mapping functions, some GTs may be used, namely, Projective, Bilinear or a simpler Affine GT, as illustrated in Fig. 1.

A. Projective Geometric Transformation

In order to simplify the mathematics used in this kind of GT, homogeneous coordinates are commonly used [35]. Thus, the Projective GT can be defined by a 3×3 matrix $\boldsymbol{H}$ verifying (2):

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

The Projective matrix $\boldsymbol{H}$ can be decomposed into three different submatrices, $\boldsymbol{L_p}$, $\boldsymbol{T_p}$ and $\boldsymbol{P_p}$:

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{L_p} & \boldsymbol{P_p} \\ \boldsymbol{T_p} & 1 \end{bmatrix}$$

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

Each submatrix is responsible for a different elementary type of GT: $\boldsymbol{T_p}$ is responsible for the description of translations, $\boldsymbol{L_p}$ is able to define linear transformations such as rotation, scaling, and shearing, and $\boldsymbol{P_p}$ describes perspective transformations.

To fully exploit the capabilities of the projective matrix $\boldsymbol{H}$, a four-point correspondence is necessary between blocks A and A'. In this case, the full transformation matrix corresponds to the following system of equations:

$$\begin{cases} x = X(u, v) = \dfrac{l_{00}u + l_{10}v + t_x}{p_x u + p_y v + 1} \\ y = Y(u, v) = \dfrac{l_{01}u + l_{11}v + t_y}{p_x u + p_y v + 1} \end{cases} \tag{4}$$

The system of equations (4) defines the necessary calculations for mapping the coordinates of every pixel of block A into the transformed block A'.

The number of available DoF is directly related with the number of known points of correspondence which exist between both images. For less than four points of correspondence, simpler transformations can be represented by the perspective model. For example, if one point is known, the only component that can be possibly described is a translation, i.e., $\boldsymbol{T_p} = [t_x, t_y]$, $\boldsymbol{P_p} = [0, 0]^T$ and $\boldsymbol{L_p} = \boldsymbol{I}$. This case is defined by (5):

$$[x, y, 1] = [uh, vh, h] \begin{bmatrix} \boldsymbol{I} & 0 \\ \boldsymbol{T_p} & 1 \end{bmatrix}, \qquad (5)$$

which can be translated into the system of equations (6):

$$\begin{cases} x = X(u) = u + t_x \\ y = Y(v) = v + t_y \end{cases}. \qquad (6)$$

B. Bilinear Geometric Transformation

The Bilinear GT is an alternative to the Projective GT, defined by a 4 × 2 matrix $\boldsymbol{B}$ verifying (7):

$$[x, y] = [uv, u, v, 1]\, \boldsymbol{B}, \qquad (7)$$

where:

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{P_b} \\ \boldsymbol{L_b} \\ \boldsymbol{T_b} \end{bmatrix} = \begin{bmatrix} p_x & p_y \\ l_{00} & l_{01} \\ l_{10} & l_{11} \\ t_x & t_y \end{bmatrix}, \qquad (8)$$

The Bilinear GT matrix $\boldsymbol{B}$ can represent similar GTs as the Projective GT, with the same number of DoF. However, it performs a non-planar transformation, which makes it more flexible. Thus, only horizontal and vertical lines, as well as equispaced points along these directions, are preserved [36]. Diagonal lines, on the other hand, are not mapped as lines but as quadratic curves. This feature is illustrated in Fig. 1, where, in the case of the Bilinear GT, points along vertical parallel lines are kept equispaced, while the points along diagonal lines are mapped onto a quadratic curve (block A_B). When the Projective GT (block A'_P) is used, points along the parallel vertical lines do not stay equispaced but points along diagonal lines are also mapped along a line. Another property of this GT, when compared to the Projective GT, is the need for simpler calculations per pixel, given by (9):

$$\begin{cases} x = X(u, v) = uvp_x + ul_{00} + vl_{10} + t_x \\ y = Y(u, v) = uvp_y + ul_{01} + vl_{11} + t_y \end{cases}. \qquad (9)$$

C. Affine Geometric Transformation

When using either the Projective or the Bilinear GT, eight DoF are available. However, a simpler case exists, which is known as the Affine GT, that is able to describe GTs up to six DoF. The Affine GT can be described as a particular case of Projective or Bilinear GTs, by using matrices $\boldsymbol{H}$ and $\boldsymbol{B}$ with $\boldsymbol{P_p^T} = [0\ \ 0]$ and $\boldsymbol{P_b} = [0\ \ 0]$, respectively. This GT only requires three points of correspondence between images, defined by (10):

$$\begin{cases} x = X(u, v) = l_{00}u + l_{10}v + t_x \\ y = Y(u, v) = l_{01}u + l_{11}v + t_y \end{cases}. \qquad (10)$$

IV. Proposed High Order Prediction Mode

This section proposes a LF image coding method, based on a high order prediction model, which is implemented as a blockwise prediction mode in HEVC. This HOP mode is added to the set of HEVC Intra prediction modes, i.e., Planar mode, DC mode and the 33 intra Directional modes.

The proposed HOP mode predicts each block by applying a GT between two quadrilaterals, the current block and a block in the reference region, the causal area of pixels already encoded. The algorithm for the proposed prediction mode can be described through the following steps:

1) *Selection of the next set of correspondence points to be evaluated*: Selection of a quadrilateral in the causal area of pixels (from a set of pre-defined cases), with corners $\{P'_n\}$, that is mapped into the block which is being predicted, with corners $\{P_n\}$ (see left side of Fig. 2);
2) *Calculation of the GT parameters*: Calculation of the transformation parameters that map the quadrilateral defined by $\{P'_n\}$ into the one defined by $\{P_n\}$;
3) *Inverse GT mapping*: Mapping of the causal quadrilateral defined by $\{P'_n\}$ to the one defined by $\{P_n\}$, using an inverse mapping procedure with the parameters calculated in the previous step, in order to compute the block prediction error; error and the estimated number of bits to transmit the GT parameters;
4) *Estimation of the GT RD cost*: Estimation of the rate-distortion (RD) cost, *J*, associated to the GT that is being evaluated, considering the computed block prediction;
5) *Repeat the above steps to find the GT with minimum RD cost:* Evaluate iteratively all the pre-defined combinations of correspondence points and choose the one that has the minimum RD cost J.
6) *Encode the HOP mode information:* The corner displacement between the quadrilateral in the causal area $\{P'_n\}$ and the block which is being predicted $\{P_n\}$ is signaled to the decoder.

The following sub-sections explain each step of the proposed HOP mode in more detail.

A. Selection of the Correspondence Points

The major challenges faced by the proposed algorithm are the computational complexity required to estimate the optimal set of GT parameters and the necessary bit-rate for transmitting this data. To tackle both problems, a rate-distortion-complexity tradeoff is defined. From Fig. 2 (left side) it can be inferred that, if all possible four-point correspondences between the prediction block and the current block to be encoded were evaluated, the number of tested transformations per block would be larger than $(2\,W^2)^4$, i.e., for a search window ($W = 128$) more than 1.15×10^{18} correspondence possibilities per block exist. To

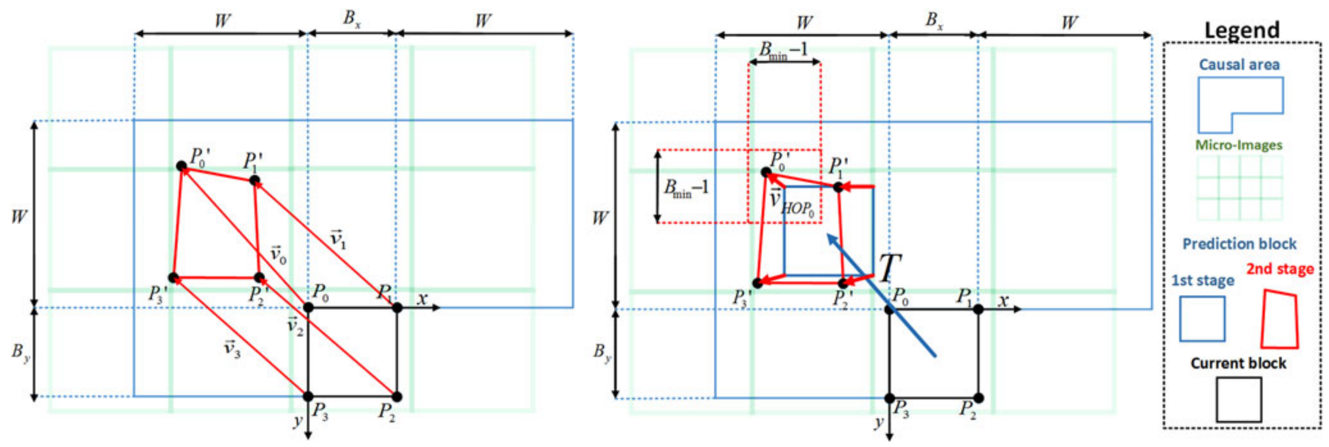
W
B_x
W
$B_{min}-1$
Legend
Causal area
Micro-Images
Prediction block
1st stage
2nd stage
Current block
P_0' P_1' P_2' P_3' P_0 P_1 P_2 P_3
$\vec{v}_0$ $\vec{v}_1$ $\vec{v}_2$ $\vec{v}_3$ $\vec{v}_{HOP_0}$
T
B_y
x
y

Fig. 2. Block prediction using a HOP model: generic single-stage HOP model mapping (left side), and proposed two-stage HOP model mapping (right side).

reduce the number of tests to a practicable number, a two-stage minimization problem is proposed, aiming to determine a good approximation to the optimal HOP model, as illustrated in Fig. 2 (right side):

1) LOP Model Estimation: In the first stage, a pure translational LOP model (two DoF) is used. The result of this stage is, the bidimensional vector, $\boldsymbol{T}$, with the lowest RD cost, pointing into the search window of the causal area (see the blue vector on the right side of Fig. 2). The search to determine $\boldsymbol{T}$ is performed using a full search algorithm, as described in [30]. The LOP estimation stage of the proposed HOP mode is based on the Self Similarity (SS) prediction method. The prediction cost is minimized by testing all the possible positions inside the search window for a single vector that relates the current block to the prediction block. The $\boldsymbol{T}$ vectors, generated by the first stage, can be either encoded explicitly, similarly to motion vectors in HEVC or using the SS-Skip mode, which creates a list of candidates that includes the $\boldsymbol{T}$ vectors used to encode neighboring blocks. If one candidate from this list is selected to encode the current block, it is only necessary to encode its index, as in the HEVC merge mode. Additionally, some predetermined vectors are added to the candidate list, referred to as MI-based candidates [30]. These candidates correspond to vectors that are very likely to be selected by the SS prediction mode, such as, vectors pointing to the same spatial position of the current MI within the left, above and above-left MIs.

2) HOP Model Estimation: In the second stage, a HOP model (up to eight DoF) is used, employing as a starting point the result of the first stage (see, respectively, the red and blue quadrilaterals on the right side of Fig. 2). For this, a set of four vectors, $\{v_{HOP_n}\}$, is computed, each of them defining the position of one corner of the reference quadrilateral, thus defining the 2D GT.

To further reduce the computational complexity of the second stage of this minimization problem, a 2D logarithmic fast search method has been adopted, which is applied to each corner of the prediction block (blue rectangle). In this case, the maximum number of search steps has been set to $\log_2(\min(B_x,\ B_y)) - 1$, depending on the size of the prediction block, i.e., B_x (width) and B_y (height). In each step, the searching points are defined

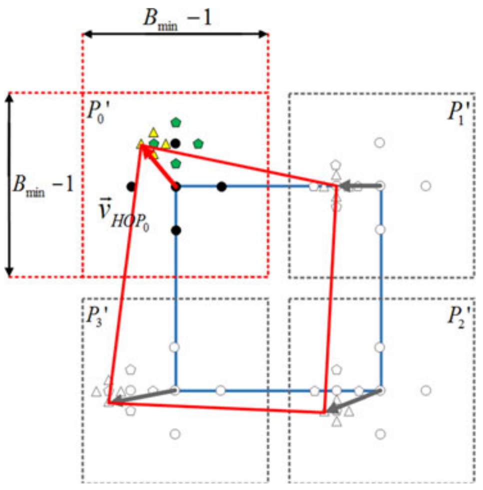

Fig. 3. Fast search method adopted for each corner of the prediction block (blue rectangle) used to estimate the HOP model (red quadrilateral).

according to a five-point small diamond-shaped basis pattern with an initial search step size equal to min$(B_x,\ B_y)/4$ [37], [38]. This 2D logarithmic fast search method using the five-point small diamond-shaped basis pattern is graphically represented in Fig. 3 across three search steps, represented, respectively, by black circles, green pentagons and yellow triangles. After each search step, the point that minimizes the RD cost function is set as the center of the next step and the search step size is halved until a unitary step value is reached. In the example of Fig. 3, in the first corner (P_0), the five points associated with the first step, represented by the black circles, are tested. The point that minimizes the RD cost function for the first search step is the black circle on the top. For the second and third search steps, the points on left, respectively, green pentagon and yellow triangle, are the points that yield the lowest RD cost. The final point is selected to define the red arrow that describes the corner displacement of the first corner of the block.

Considering that the search procedure must be applied to all the corners of the prediction block over several search steps, there are two ways of implementing this second stage search: by jointly optimizing each step of the search procedure for the four corners or by independently optimizing each step of the search procedure. By considering five points for each of the S search steps of the 2D logarithm search, the required number of search points for each option is given by $(5 \times S)^n$ or $5^n \times S$, where n is the number of corners. In order to reduce the computational complexity, the second option was used, where each step is optimized individually.

The stop condition for this search method is met when the corner step size reaches the unit. Therefore, the example shown in Fig. 3 represents the unitary steps as the yellow triangles. Since the underlying codec uses variable block sizes, S will depend on the block size. The search window for each corner is limited to $\min(B_x, B_y) - 1$, as illustrated in Fig. 3 (see the dashed red block).

The quadrilateral used by the HOP model estimation may be scaled to increase pixel precision. Figs. 2 and 3 illustrate the second stage applied to a blue rectangle with the same size of the block being predicted (in black) to not overload the figures. However, in our implementation, a rectangle, twice the size of the original block, is used to determine the HOP model. This modification means that an integer pixel displacement in one of the corners of the large quadrilateral corresponds to a sub-pixel displacement in the area of the original rectangle. For a block twice the size of the original block, one extra search step is performed by the 2D logarithmic search algorithm that is used at the HOP stage. This occurs because the stopping condition for the search algorithm is the unitary step size. The pixel precision can be further extended by using a rectangle with sides four or eight times the size of the original blue rectangle, which increase, the number of search steps by one or two, respectively. After extensive testing, the best solution in a RD sense was adopted, that is increasing the blue rectangle to twice the original size, despite requiring one extra step.

As the second stage of the HOP search can be biased by the first stage result, the global result of the search method also tests the $\boldsymbol{T}$ vectors used in the previously encoded neighboring blocks (vector predictors), instead of considering only the best $\boldsymbol{T}$ vector from the first stage. Additionally, other $\boldsymbol{T}$ vectors can be tested in conjunction with the HOP model estimation, e.g. top ten candidates from the first stage. However, it was experimentally verified by the authors that the vectors that are more RD cost efficient are the $\boldsymbol{T}$ vector predictors.

The proposed approach can be implemented using either the Projective GT defined in (3):

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \boldsymbol{T} & 0 \end{bmatrix} + \begin{bmatrix} \boldsymbol{L'_p} & \boldsymbol{P'_p} \\ \boldsymbol{T'_p} & 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{L'_p} & \boldsymbol{P'_p} \\ \boldsymbol{T} + \boldsymbol{T'_p} & 1 \end{bmatrix} \quad (11)$$

or the Bilinear GT defined in (8):

$$\boldsymbol{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \boldsymbol{T} \end{bmatrix} + \begin{bmatrix} \boldsymbol{P'_b} \\ \boldsymbol{L'_b} \\ \boldsymbol{T'_b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{P'_b} \\ \boldsymbol{L'_b} \\ \boldsymbol{T} + \boldsymbol{T'_b} \end{bmatrix}. \quad (12)$$

Where $\boldsymbol{T}$ is the vector estimated during the LOP stage and $\boldsymbol{T'}$, $\boldsymbol{L'}$ and $\boldsymbol{P'}$ are the GT parameters that describe the HOP stage.

B. Calculation of the GT Parameters

After obtaining vector $\boldsymbol{T}$ (see the right side of Fig. 2), it is possible to determine submatrices $\boldsymbol{T'}$, $\boldsymbol{P'}$ and $\boldsymbol{L'}$ in (11) and (12), by using their width and height, B_x and B_y, respectively, and the small vectors associated with the corner position change of the blue rectangle:

$$\begin{cases} \vec{v}_{HOP_0} = (u_0, v_0) \\ \vec{v}_{HOP_1} = (u_1 - (B_x - 1), v_1) \\ \vec{v}_{HOP_2} = (u_2 - (B_x - 1), v_2 - (B_y - 1)) \\ \vec{v}_{HOP_3} = (u_3, v_3 - (B_y - 1)) \end{cases} \quad (13)$$

Note that in the proposed two-stage approach, vectors $\vec{v}_n$, represented in the left of Fig. 2, correspond to the sum of vector $\boldsymbol{T}$ from the first stage, with the four smaller vectors from the second stage, i.e., $\vec{v}_n = T + \vec{v}_{HOP_n}$, represented in the right of Fig. 2. If the Projective GT is used some auxiliary variables are defined:

$$\begin{cases} \Delta u_1 = u_1 - u_2 \\ \Delta u_2 = u_3 - u_2 \\ \Delta u_3 = u_0 - u_1 + u_2 - u_3 \end{cases}$$

$$\begin{cases} \Delta v_1 = v_1 - v_2 \\ \Delta v_2 = v_3 - v_2 \\ \Delta v_3 = v_0 - v_1 + v_2 - v_3 \end{cases}. \quad (14)$$

The Affine GT can be defined by any three of the four vectors ($\vec{v}_{HOP}$). In this paper, the first three vectors,$\vec{v}_{HOP_0}$,$\vec{v}_{HOP_1}$ and $\vec{v}_{HOP_2}$ are generated using the second stage of the proposed approach, where the remaining vector, $\vec{v}_{HOP_3}$, is calculated assuming $\Delta u_3 = \Delta v_3 = 0$, thus resulting in $\vec{v}_{HOP_3} = (u_0 - u_1 + u_2, v_0 - v_1 + v_2 - (B_y - 1))$.Using (14), the individual parameters in the submatrices can then be calculated by (15) for the Projective GT:

$$\boldsymbol{P'_p} = \begin{bmatrix} \dfrac{1}{B_x - 1} \dfrac{\begin{vmatrix} \Delta u_3 & \Delta u_2 \\ \Delta v_3 & \Delta v_2 \end{vmatrix}}{\begin{vmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{vmatrix}} \\ \dfrac{1}{B_y - 1} \dfrac{\begin{vmatrix} \Delta u_1 & \Delta u_3 \\ \Delta v_1 & \Delta v_3 \end{vmatrix}}{\begin{vmatrix} \Delta u_1 & \Delta u_2 \\ \Delta v_1 & \Delta v_2 \end{vmatrix}} \end{bmatrix} \text{ and}$$

$$\boldsymbol{L'_p} = \begin{bmatrix} \frac{u_1 - u_0}{B_x - 1} + p_x u_1 & \frac{u_3 - u_0}{B_y - 1} + p_y u_3 \\ \frac{v_1 - v_0}{B_x - 1} + p_x v_1 & \frac{v_3 - v_0}{B_y - 1} + p_y v_3 \end{bmatrix}. \quad (15)$$

Similarly, for the Bilinear GT, the corresponding submatrices are calculated by (16):

$$\boldsymbol{P'^T_b} = \begin{bmatrix} \frac{u_0 - u_1 + u_2 - u_3}{(B_x - 1)(B_y - 1)} \\ \frac{v_0 - v_1 + v_2 - v_3}{(B_x - 1)(B_y - 1)} \end{bmatrix} \text{ and}$$

$$\boldsymbol{L'_b} = \begin{bmatrix} \frac{u_3 - u_0}{B_y - 1} & \frac{v_3 - v_0}{B_y - 1} \\ \frac{u_1 - u_0}{B_x - 1} & \frac{v_1 - v_0}{B_x - 1} \end{bmatrix}. \quad (16)$$

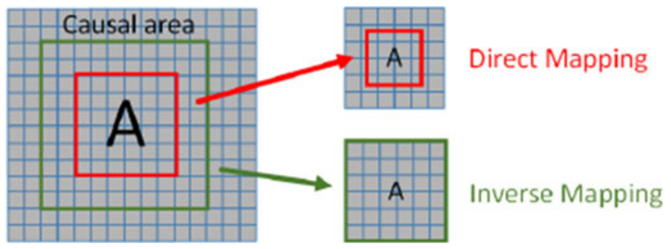

Fig. 4. Example of direct mapping and inverse mapping when a scale GT is applied.

For both cases we have:

$$\boldsymbol{T}'_{\boldsymbol{p}} = \boldsymbol{T}'_{\boldsymbol{b}} = \begin{bmatrix} u_0 & v_0 \end{bmatrix}. \tag{17}$$

C. Inverse GT Mapping

As previously mentioned, a GT between two blocks corresponds to a mapping of every pixel within one block into the other block, e.g., the mapping functions (4) and (9) correspond to the Projective and Bilinear GT, respectively. When the mapping is performed from the rectangular block to be encoded to an arbitrary reference quadrilateral it is called a direct mapping, otherwise it is called an inverse mapping. An example of how both mapping procedures for a simple scaling GT can be found in Fig. 4. As can be observed in Fig. 4, when direct mapping is used the final quadrilateral shape (red block) does not match the desired reference block pixel grid, requiring to perform pixel interpolation prior to calculate the distortion between the transformed block and the reference block. For the sake of simplicity, an inverse mapping has been adopted, as it generates a rectangular prediction block with the same dimensions of the block to be encoded.

Thus, regardless of the size of the quadrilateral used for estimation, (4) and (9) take as input the coordinates of the block to be encoded, i.e., $u \in [0, B_x - 1]$ and $v \in [0, B_y - 1]$, and generate as output the coordinates (x, y) in the causal area, where the reference pixel value is going to be extracted from. Since x and y are typically fractional values, a bilinear interpolation filter is used to compute the actual pixel value.

D. Estimation of the GT RD Cost

The optimal HOP model for each block is determined through RD optimization, minimizing the associated Lagrangian cost, $J = D + \lambda R$, over the entire set of pre-defined GT. D refers to the distortion between the prediction block and the current block, R is the estimated number of bits used to encode the block using the GT under evaluation, and λ is the Lagrange multiplier, computed as in HM version 15.0 for Intra-coded frames. The parameter λ is the same for all prediction modes, including the intra modes, so no biases in terms of prediction mode selection are introduced. In this paper, D, is computed as the sum of absolute differences (SAD) in the pixel domain, in the first stage, and SAD in the Hadamard domain, in the second stage, as suggested in [39].

By using a two-stage method it is possible to evaluate if it is more advantageous to use LOP or HOP for each block, by comparing the associated costs, given by:

$$J_{LOP} = D_{LOP} + \lambda R_{LOP} \text{ and}$$
$$J_{HOP} = D_{HOP} + \lambda R_{HOP}, \tag{18}$$

where R_{LOP} and R_{HOP} are the estimated number of bits for the corresponding coding mode.

The usage of LOP or HOP is conveyed to the decoder through a binary flag, F_{HOP}. When LOP is considered more efficient in a RD sense, $F_{HOP} = 0$, and only $\boldsymbol{T}$ is transmitted in the bitstream. On the contrary, if HOP is used, all the elements that describe $\boldsymbol{T}$ are transmitted, followed by $F_{HOP} = 1$ and the four additional $\vec{v}_{HOP_n}$ vectors.

The number of bits required to signal the HOP mode, R_{HOP}, is the sum of R_{LOP} and the estimated bits for encoding the four vectors, $\vec{v}_{HOP_n}$, that define the used HOP model. The rate of these small amplitude vectors is estimated using the same procedure as vector $\boldsymbol{T}$.

E. Encode the HOP Mode Information

After finding the optimal HOP model, the cost of the HOP mode, J_{HOP}, is compared against 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 encoded. For this, the context adaptive binary arithmetic coding (CABAC) entropy coding method used by HEVC is used to encode the HOP mode information. The CABAC entropy coder is based on three steps: (i) binarization of syntax elements, (ii) context modeling, and (iii) binary arithmetic coding. In this implementation, these three steps have been maintained using, however, new contexts. Vectors $\boldsymbol{T}$ and $\vec{v}_{HOP_n}$, and lag, F_{HOP}, are transmitted to the decoder using the HM approach for motion vectors and merge flags [40].

To encode $\boldsymbol{T}$, the same syntax elements of HEVC for motion data are used, i.e., motion vector differences, MVP index, reference picture list (RPL) and RPL index.

The way the HOP model information is conveyed to the decoder can highly influence the coding efficiency. One possible approach is to send the GT parameters, i.e., in the $\boldsymbol{H}$ or $\boldsymbol{B}$ matrix, which need to be represented with high precision. Alternatively, as proposed in this paper, the encoder just sends the four vectors, $\vec{v}_{HOP_n}$, which can be represented with just a few bits. The major advantage of encoding the GT parameters matrix is that they do not need to be recalculated at the decoder side through (13)–(17). However, they need to be encoded with a very high precision because these values are not very robust to quantization [11]. Consequently, encoding the vectors, $\vec{\boldsymbol{v}}_{\boldsymbol{HOP_n}}$, leads to higher compression efficiency.

V. Experimental Results

In this section the performance of the proposed lenslet LF coding solution, incorporating the HOP mode, is evaluated in comparison with state-of-the-art coding solutions based on LOP approaches. First, this section describes the test conditions, including the used lenslet LF test images, the benchmark solutions and the relevant test parameters. Afterwards, experimental results comparing the RD performance of different types

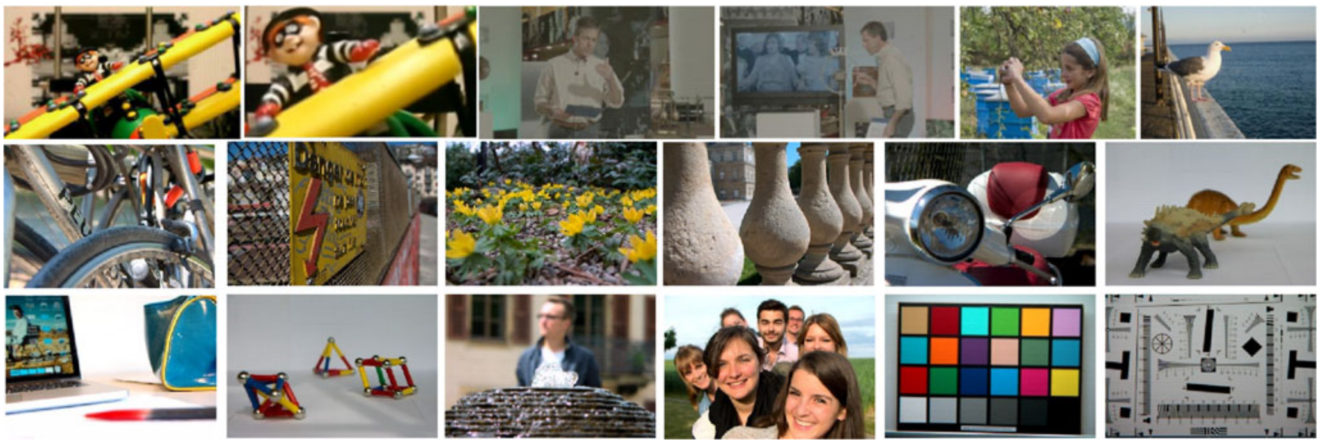

Fig. 5. LF test images part of the experimental test setup. First row (from left to right): Plane and Toy (frame 0 and 150), Demichelis Spark (frame 0), Demichelis Cut (frame 0), Laura and Seagull. Second and third rows: sub-set of the LF EPFL dataset.

of prediction models are presented and discussed. These results are complemented with some statistical information about prediction mode usage and an evaluation of the quality of the rendered views, as proposed in [23], using the coded LF images.

A. Test Conditions

In order to evaluate the RD performance of the proposed LF coding solution, two types of LF images were selected for the experimental test setup. The first type of images were acquired using LF cameras with a focused (FOC) optical setup [41], [42]. The second type of images were acquired using a Lytro Illum camera that is commercially available and uses an unfocused (UNF) optical setup. This second set of images constitutes the dataset used for the 2016 ICME Grand Challenge on LF image compression extracted from the EPFL dataset [23]. The central rendered views of all the test images are shown in Fig. 5, where the first row corresponds to the first type of images and the second and third rows correspond to the second one. This selection includes LF images with different resolutions, MI resolutions and types of microlens arrays, with different MI shape. *Plane and Toy* images have a resolution of 1920 $\times$ 1088 (MIs 28 $\times$ 28); *Demichelis* images have a resolution of 2880 $\times$ 1620 (MIs 38 $\times$ 38); *Laura* and *Seagull* have a resolution of 7240 $\times$ 5432 (MIs 75 $\times$ 75); EPFL images have a resolution of 7728 $\times$ 5368 pixels (MIs 15 $\times$ 15).

The proposed HOP mode was implemented into the HEVC test model version 15.0 (HM 15.0) as an additional intra prediction mode. This LF codec, corresponding to the proposed solution, will be referred to as HEVC-HOP, where HEVC using only the standard Intra modes is simply referred to as HEVC. Additionally, the work in [30] is used as benchmark for RD performance and it is referred to as HEVC-SS.

The common HM test conditions were adopted, using QP values of 22, 27, 32 and 37. The causal window size W is 128 for both HEVC-HOP and HEVC-SS, for every encoded image. The number of available SS or $\boldsymbol{T}$ vector predictors, used for coding, is 2. These $\boldsymbol{T}$ vector predictors are used as additional vector $\boldsymbol{T}$ candidates for the LOP model estimation stage. As mentioned in the previous section, these alternative vectors are tested in order to have a more unbiased result when estimating the HOP model. The number of candidates available for SS-Skip is 5 in both HEVC-SS and HEVC-HOP.

B. Experimental Results

All the LF images in Fig. 5 are encoded and decoded using the HEVC, HEVC-HOP and HEVC-SS codecs, and the RD performance is evaluated using a Bjøntegaard Delta Metric. Additionally, several variants of HEVC-HOP are tested. These variants of HEVC-HOP are referred to as HEVC-HOP-A, HEVC-HOP-P and HEVC-HOP-B, respectively for Affine (six DoF), Projective (eight DoF) and Bilinear (eight DoF) GTs. Table I shows the RD performance comparison between HEVC and HEVC-SS, and between HEVC-SS and the various HEVC-HOP variants.

1) Comparison Between LOP and HOP: Table I shows that HEVC-SS can outperform HEVC, for all tests, with bitrate savings up to 45.35%. Nevertheless, all versions of the proposed HEVC-HOP method are even more efficient than HEVC-SS to encode LF images. This increased performance, with bitrate savings up to 12.62% for certain images relatively to HEVC-SS (49.82% relatively to HEVC), comes from the use of a higher order prediction model. Since HEVC-SS is limited to two DoF, it is not able to accurately describe block transformations more complex than a simple translation. When comparing the results by means of comparing the effectiveness of adding prediction tools with more than two DoF, it is possible to notice that for the encoded LF images, the best case is when eight DoF are used. If eight DoF are available, i.e., when HEVC-HOP-P is being used, four points of correspondence are transmitted, which allows the description of not only translations, but also rotations, scaling, shearing and perspective changes. In this case, although extra information needs to be encoded, relative to the HEVC-SS case, the bitrate savings increases to 5.86% (28.81% relative to HEVC), in average, for all tested LF images.

2) Comparison Between the Proposed GTs: The proposed prediction mode HEVC-HOP-B, using a Bilinear GT, can achieve similar results to HEVC-HOP-P for most images, both

TABLE I
BD-PSNR-Y AND BD-RATE RESULTS COMPARING HEVC, HEVC-SS (TWO DOF) AND HEVC-HOP, USING SIX DOF AND EIGHT DOF AND TWO DIFFERENT KINDS OF GEOMETRIC TRANSFORMATIONS

Image	HEVC-SS (2 DoF) vs HEVC		HEVC-HOP-A (6 DoF) vs HEVC-SS (2 DoF)		HEVC-HOP-P (8 DoF) vs HEVC-SS (2 DoF)		HEVC-HOP-B (8 DoF) vs HEVC-SS (2 DoF)	
	BD-PSNR-Y	BD-RATE	BD-PSNR-Y	BD-RATE	BD-PSNR-Y	BD-RATE	BD-PSNR-Y	BD-RATE
Plane and Toy, frame 0 (PT0)	0.90 dB	−14.64%	0.23 dB	−3.93%	**0.27 dB**	**−4.66%**	0.23 dB	−4.02%
Plane and Toy, frame 150 (PT150)	1.44 dB	−19.02%	0.64 dB	−9.44%	**0.75 dB**	**−11.05%**	0.71 dB	−10.50%
Demichelis Spark, frame 0 (DS)	1.09 dB	−31.43%	0.23 dB	−7.41%	**0.26 dB**	**−8.39%**	0.26 dB	−8.31%
Demichelis Cut, frame 0 (DC)	1.05 dB	−29.25%	0.26 dB	−8.06%	**0.30 dB**	**−9.14%**	0.29 dB	−8.83%
Laura (LAURA)	2.26 dB	−30.35%	0.15 dB	−2.76%	0.27 dB	−4.78%	**0.32 dB**	**−5.62%**
Seagull (SEAGULL)	2.81 dB	−42.78%	0.22 dB	−4.89%	0.31 dB	−6.82%	**0.43 dB**	**−9.21%**
Bikes (BIKES)	0.81 dB	−18.50%	0.11 dB	−2.91%	**0.13 dB**	**−3.34%**	0.11 dB	−2.97%
Danger de Mort (DANGER)	0.61 dB	−14.67%	0.10 dB	−2.72%	**0.11 dB**	**−2.94%**	0.10 dB	−2.75%
Flowers (FLOWERS)	0.17 dB	−4.10%	0.03 dB	−0.75%	**0.03 dB**	**−0.77%**	0.03 dB	−0.67%
Stone Pillars (STONE)	0.25 dB	−6.31%	**0.02 dB**	**−0.61%**	0.02 dB	−0.57%	0.02 dB	−0.57%
Vespa (VESPA)	0.89 dB	−28.73%	0.10 dB	−3.90%	**0.14 dB**	**−5.30%**	0.12 dB	−4.89%
Ankylosaurus & Diplodocus (ANKY)	1.43 dB	−45.35%	0.05 dB	−1.26%	**0.06 dB**	**−4.46%**	0.05 dB	−2.91%
Desktop (DESKTOP)	0.48 dB	−13.73%	0.26 dB	−8.05%	0.25 dB	−7.78%	**0.29 dB**	**−8.93%**
Magnets (MAGNTES)	0.66 dB	−22.95%	0.06 dB	−3.52%	**0.06 dB**	**−5.19%**	0.05 dB	−2.62%
Fountain & Vincent (FOUNTAIN)	1.49 dB	−30.72%	0.17 dB	−4.39%	**0.20 dB**	**−5.31%**	0.20 dB	−5.28%
Friends (FRIENDS)	0.22 dB	−8.10%	0.05 dB	−2.01%	0.06 dB	−2.21%	**0.06 dB**	**−2.36%**
Color chart (COLOR)	1.49 dB	−42.84%	0.27 dB	−11.19%	0.31 dB	−12.30%	**0.32 dB**	**−12.62%**
ISO Chart (ISO)	1.42 dB	−41.35%	0.24 dB	−9.28%	0.27 dB	−10.42%	**0.27 dB**	**−10.47%**
AVG. FOC	1.59 dB	−27.91%	0.29 dB	−6.08%	0.36 dB	−7.47%	**0.37 dB**	**−7.75%**
AVG. UNF	0.83 dB	−23.11%	0.12 dB	−4.22%	**0.14** dB	**−5.06%**	0.14 dB	−4.75%
AVG. ALL	1.08 dB	−24.71%	0.18 dB	−4.84%	**0.21** dB	**−5.86%**	0.21 dB	−5.75%

in terms of average PSNR (BD-PSNR) and bitrate savings. However, comparing the average performance of each method regarding the type of camera models (AVG. FOC and AVG. UNF) it is possible to observe that HEVC-HOP-P is slightly more efficient for the unfocussed model images and HEVC-HOP-B is slightly more efficient for the focused model images. In the case of HEVC-HOP-A only six DoF are available because only three points of correspondence are transmitted. When compared HEVC-HOP-A to HEVC-HOP-P, the bitrate savings gains relatively to HEVC-SS are reduced to 4.84% (28.12% relatively to HEVC) on average considering all tested LF images, which may be due to the fact that HEVC-HOP-A is not able to compensate for perspective changes. However, in terms of computational complexity HEVC-HOP-A is approximately 4.5 times faster than HEVC-HOP-P. Note that none of the implementations is optimized in terms of computational complexity; therefore, the reported values for comparison may vary. It is worthwhile mentioning that for some cases (e.g., STONE test image) HEVC-HOP-A can outperform both eight DoF GTs. In HEVC-HOP-P four correspondence points are always encoded, even if only three are necessary. Since in some cases, more information might be transmitted to describe the same GT, HEVC-HOP-P is, for this particular test image, less efficient than HEVC-HOP-A.

Regarding the computational complexity, a study was performed using the image VESPA, from the EPFL dataset of LF images. This image was encoded and decoded using the codecs, HEVC, HEVC-SS, HEVC-HOP-A, HEVC-HOP-P and HEVC-HOP-B, with QP = 32. These tests were performed using a PC equipped with an Intel Xeon CPU E3-1240 Vand 24GB of RAM, running Ubuntu 16.04. The obtained running time to encode and decode each image is depicted in Table IV. The computational complexity of the proposed schemes must be compared to HEVC-SS, as it is used as reference. The HEVC-SS complexity is equivalent to encoding a P-Slice in HEVC [30]. As can be seen from Table IV, the proposed algorithm increases the computational burden at the encoder side, where HEVC-HOP-A, HEVC-HOP-P and HEVC-HOP-B are 1.51, 6.84 and 5.23 times more complex than HEVC-SS, respectively. However, at the decoder the running time is reduced in relation to HEVC-SS. As the proposed method uses a more efficient prediction, the LF image encoder creates a lower number of partitions than the HEVC-SS.

One of the most important advantages of the proposed prediction method is the ability to choose between the LOP stage and the HOP stage for each image block. This decision is taken based on RDO criteria, which allows the proposed HEVC-HOP to outperform HEVC-SS in all cases. In Table II it is possible to observe that despite the HOP stage of the proposed prediction method being used more frequently than the LOP stage, there is always a considerable part of the image that is encoded using only the LOP prediction mode. However, the fact that the HOP stage is used more often than the LOP stage alone indicates that, although the proposed method HEVC-HOP-P requires additional overhead for transmitting the prediction information, i.e., one flag (F_{HOP}) and four vectors ($\{\vec{v}_{HOP_n}\}$), it is very efficient in reducing the distortion between the current block and prediction block, therefore reducing the RD cost. An example of this can be seen in Fig. 6 where a comparison between the generated prediction block using either the LOP method or the proposed HOP method is shown. In this example, the prediction block generated using the

TABLE II
AVERAGE PREDICTION MODE USAGE ACROSS THE FOUR QPs, IN PERCENTAGE OF PIXELS, FOR THE HEVC-HOP-P CASE

Image	DC, Planar and Directional	Proposed prediction method		SS-Skip [30]
		LOP stage	HOP stage	
PT0	**57.76%**	3.32%	22.81%	16.12%
PT150	26.53%	4.39%	**51.27%**	17.62%
DS	27.35%	3.84%	**40.31%**	28.51%
DC	27.25%	1.93%	**44.04%**	26.78%
LAURA	21.51%	10.25%	**45.22%**	23.03%
SEAGULL	13.87%	9.27%	**41.81%**	35.05%
BIKES	**44.12%**	5.69%	33.44%	16.75%
DANGER	**49.28%**	5.99%	31.15%	13.59%
FLOWERS	**81.54%**	2.63%	10.55%	5.28%
STONE	**82.66%**	2.75%	9.60%	4.99%
VESPA	**38.42%**	7.94%	30.03%	23.61%
ANKY	24.39%	10.89%	30.26%	**34.46%**
DESKTOP	**57.50%**	2.67%	15.68%	24.14%
MAGNETS	28.88%	9.57%	28.42%	**33.14%**
FOUNTAIN	30.12%	8.38%	**37.66%**	23.84%
FRIENDS	**78.01%**	3.08%	11.35%	7.57%
COLOR	15.62%	8.94%	**38.14%**	37.30%
ISO	11.40%	11.29%	**41.57%**	35.75%

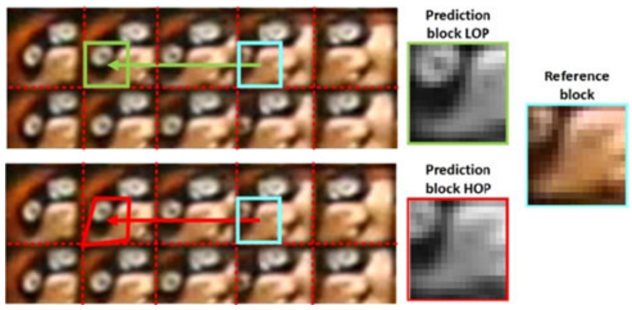

Fig. 6. Comparison between the prediction block generated by LOP and HOP stages.

proposed HOP stage has a lower RD cost ($J_{HOP} = 5080$) than the prediction block using LOP ($J_{LOP} = 7851$), despite the extra bits necessary to convey to the decoder the GT parameters ($\{\vec{v}_{HOP_n}\} = \{(3;0),(4;0),(2;-1),(2;1)\}$).

3) Experimental Results for Rendered SAIs: To further evaluate the compression efficiency of the proposed HEVC-HOP-P, the objective quality of the SAIs extracted from the encoded LF images was tested. The experimental methodology adopted in [23] for the Lytro Illum unfocused camera setup was used. In this case, the PSNR-Y is calculated as an average of the PSNR-Y of 13 $\times$ 13 SAIs. This average PSNR-Y compares reference and reconstructed (i.e., encoded and decoded) SAIs. The processing chain designed to generate the SAIs consists in converting the hexagonal lenslet LF image to a square lenslet LF image (with 15x15 pixels per MI) and then, extracting one pixel, in a fixed position from each square MI, to render each SAI. For this case only the images from [23] have been used. The only difference to the methodology in [23] is that, instead of using fixed compression ratios, the results are calculated using the reconstructed images attained with fixed QPs (22, 27, 32 and 37), i.e., the number of bits is the same as in the previously used methodology.

TABLE III
BD-PSNR-Y AND BD-RATE RESULTS COMPARING HEVC, HEVC-SS (2 DoF) AND HEVC-HOP-P (8 DoF) USING THE TESTING METHODOLOGY OF [23]

Image	HEVC-SS vs HEVC		HEVC-HOP-P vs HEVC-SS	
	BD-PSNR-Y	BD-RATE	BD-PSNR-Y	BD-RATE
BIKES	0.78 dB	−20.83%	0.08 dB	−2.41%
DANGER	0.57 dB	−16.08%	0.08 dB	−2.41%
FLOWERS	0.17 dB	−4.67%	0.02 dB	−0.69%
STONE	0.27 dB	−8.43%	−0.03 dB	0.95%
VESPA	0.66 dB	−32.79%	0.09 dB	−5.34%
ANKY	1.48 dB	−62.57%	0.11 dB	−6.87%
DESKTOP	0.33 dB	−15.23%	0.10 dB	−5.19%
MAGNETS	0.64 dB	−37.90%	0.08 dB	−5.97%
FOUNTAIN	1.22 dB	−35.38%	0.12 dB	−4.34%
FRIENDS	0.15 dB	−10.40%	0.03 dB	−1.72%
COLOR	1.26 dB	−58.40%	0.18 dB	−12.86%
ISO	1.49 dB	−45.39%	0.26 dB	−10.81%
AVG.	0.75 dB	−29.01%	0.09 dB	−4.81%

TABLE IV
CODEC COMPUTATIONAL COMPLEXITY COMPARISON

Encoder	HEVC	HEVC-SS	HEVC-HOP-A	HEVC-HOP-P	HEVC-HOP-B
Run time (h)	0.06	3.88	5.88	26.54	20.30
vs HEVC-SS	0.02	1	1.51	6.84	5.23
Decoder	HEVC	HEVC-SS	HEVC-HOP-A	HEVC-HOP-P	HEVC-HOP-B
Run time (s)	1.70	33.56	30.54	29.98	28.56
vs HEVC-SS	0.05	1	0.91	0.89	0.85

From the results presented in Table III it is possible to observe that there is a coherence between the results for the encoded LF images and the rendered SAIs. The average bitrate savings achieved by HEVC-HOP-P relative to HEVC-SS are very similar for both cases. Nevertheless, as the RD cost was not optimized on each SAI, the results are not exactly the same. As previously explained, when using the HEVC-HOP-P method the LF image is encoded "as is", without the need to know any information about the used lenslet based LF camera. This may explain the bitrate increase for image STONE, as the proposed method calculates the RD cost based on the lenslet LF image, instead of the generated view or SAI.

Additionally, a comparison between the proposed HEVC-HOP-P and the state-of-the-art method [24] was performed. In [24], a comparison in relation to JPEG, using the EPFL dataset, reports a gain of 4.54 dB in the BD-PSNR. Similarly, this image set were encoded by the proposed HEVC-HOP-P, with QPs {22, 27, 32, 37}, achieving a BD-PSNR gain of 4.83 dB, in relation to JPEG. For a wider QP range {17, 22, 27, 32, 37}, the gain of HEVC-HOP-P decreases to 4.34 dB. Thus, it is fair to assume that the proposed HEVC-HOP-P has a very similar performance to the state-of-the-art method [24].

4) Results for Different Lenslet LF Camera Models: In general, the bitrate savings across the different codecs when compared to HEVC are higher when encoding LF images captured with cameras using a focused LF camera model. This is possible to see when comparing the average bitrate savings for the

LF images captured with an unfocused camera model or the focused camera model in Table II. This occurs because HEVC-SS and HEVC-HOP are based on matching prediction tools. In the focused images, the MIs are focused, therefore sharper than the unfocussed images. In sharper MI, more prominent features exist and therefore the block matching is more reliable [34]. Additionally, since the incident light in the camera's sensor in the unfocused case is focused at infinity, the disparity between MIs tends to be zero, which means that theoretically no perspective compensation can be matched. This can be justified by the noticeable lower relative prediction mode usage, shown in Table II, for the proposed prediction method as well as SS-Skip for most LF images captured with unfocused camera models.

The proposed HOP model is also more suited to adapt to non-rectangular shape MIs, e.g., hexagonal and circular shape, when compared to LOP model based methods. This happens because the corners of the prediction blocks, when using the proposed HOP model, are flexible to adapt for different block shapes.

VI. Conclusion

In this paper, a HOP mode for LF image coding was proposed, using geometric transformations of up to eight DoF. The proposed HOP mode is a two-stage block-wise approach that is able to achieve RD efficiency gains, relative to a LOP state-of-the-art solution for LF image coding and HEVC. These gains occur, regardless of the LF camera model, MI and LF image resolution and microlens array type. Experimental results show average bitrate savings of 5.86% and 28.81%, when compared to a LOP state-of-the-art solution and HEVC, respectively, across different types of LF images, when using the Projective GT. It is also possible to conclude that, the GTs with eight DoF, namely Projective and Bilinear, are generally more efficient than Affine GT in a RD sense.

An additional testing methodology, based on the SAI objective quality, was also used to confirm the bitrate savings when comparing HOP with LOP tools. In this case the average bitrate savings achieved is 4.81% and 31.77% when comparing the HOP mode with the state-of-the-art LOP solution and HEVC, respectively.

Future work will include the investigation of GT parameter prediction techniques and optimal HOP model selection, aiming to combine them in the same codec. Additionally, entropy encoding improvements will also be considered, namely, the binarization of the HOP model vectors.

References

[1] T. Georgiev and A. Lumsdaine, "Rich image capture with plenoptic cameras," in *Proc. IEEE Int. Conf. Comput. Photography*, Cluj-Napoca, Romania, 2010, pp. 1–8.

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

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

[4] A. Lumsdaine and T. Georgiev, "The focused plenoptic camera," in *Proc. IEEE Int. Conf. Comput. Photography*, 2009, pp. 1–8.

[5] 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.

[6] "JPEG PLENO abstract and executive summary," ISO/IEC JTC 1/SC 29/WG1 N6922, Sydney, NSW, Australia, Feb. 2015.

[7] S. Kumar and R. C. Jain, "Low complexity fractal-based image compression technique," *IEEE Trans. Consum. Electron.*, vol. 43, no. 4, pp. 987–993, Nov. 1997.

[8] H. Huang, J. W. Woods, Y. Zhao, and H. Bai, "Control-point representation and differential coding affine-motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 10, pp. 1651–1660, Oct. 2013.

[9] T. Wiegand, E. Steinbach, and B. Girod, "Affine multipicture motion-compensated prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 197–209, Feb. 2005.

[10] M. Tok, M. Esche, A. Glantz, A. Krutz, and T. Sikora, "A parametric merge candidate for high efficiency video coding," in *Proc. Data Compression Conf.*, 2013, pp. 33–42

[11] A. Krutz, A. Glantz, M. Tok, M. Esche, and T. Sikora, "Adaptive global motion temporal filtering for high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1802–1812, Dec. 2012.

[12] M. Tok, A. Glantz, A. Krutz, and T. Sikora, "Feature-based global motion estimation using the Helmholtz principle," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2011, pp. 1561–1564.

[13] R. C. Kordasiewicz, M. D. Gallant, and S. Shirani, "Affine motion prediction based on translational motion vectors," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1388–1394, Oct. 2007.

[14] X. San, H. Cai, J. G. Lou, and J. Li, "Multiview image coding based on geometric prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 11, pp. 1536–1548, Nov. 2007.

[15] K. Müller *et al.*, "3D high-efficiency video coding for multi-view video and depth data," *IEEE Trans. Image Process.*, vol. 22, no. 9, pp. 3366–3378, Sep. 2013.

[16] R. J. S. Monteiro, N. M. M. Rodrigues, and S. M. M. Faria, "Disparity compensation using geometric transforms," in *Proc. 3DTV-Conf., True Vis. Capture, Transmiss. Display 3D Video*, 2014, pp. 1–4.

[17] R. J. S. Monteiro, N. M. M. Rodrigues, and S. M. M. Faria, "Geometric transforms and reference picture list optimization for efficient disparity compensation," in *Proc. Int. Conf. Image Process. Theory, Tools Appl.*, 2015, pp. 370–375.

[18] A. Aggoun, "A 3D DCT compression algorithm for omnidirectional integral images," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, vol. 2, pp. II–II.

[19] M. C. Forman and A. Aggoun, "Quantisation strategies for 3D-DCT-based compression of full parallax 3D images," in *Proc. Int. Conf. Image Process. Its Appl.*, 1997, vol. 1, pp. 32–35.

[20] A. Aggoun, "Compression of 3D integral images using 3D wavelet transform," *J. Display Technol.*, vol. 7, no. 11, pp. 586–592, Nov. 2011.

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

[22] A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao, "Data formats for high efficiency coding of Lytro-Illum light fields," in *Proc. Int. Conf. Image Process. Theory, Tools Appl.*, 2015, pp. 494–497.

[23] "ICME 2016 grand challenge: Light-field image compression." [Online]. Available: http://mmspg.epfl.ch/files/content/sites/mmspl/files/shared/LF-GC/CFP.pdf, Accessed on: Dec. 1, 2016.

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

[25] J. Boyce, J. Chen, E. Alshina, G. J. Sullivan, and J.-R. Ohm, "Algorithm description of Joint Exploration Test Model 3 (JEM3)," ISO/IEC JTC1/SC29/WG11/N16276, Jun. 2016.

[26] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.

[27] B. Girod, C.-L. Chang, P. Ramanathan, and X. Zhu, "Light field compression using disparity-compensated lifting," in *Proc. Int. Conf. Multimedia Expo*, 2003, vol. 1, pp. 373–376.

[28] Y. Li, M. Sjöström, and R. Olsson, "Coding of plenoptic images by using a sparse set and disparities," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2015, pp. 1–6.

[29] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, Jan. 2016.

[30] 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.

[31] C. Conti, P. Nunes, and L. D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2016, pp. 1–4.

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

[33] 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.

[34] Y. Li, R. Olsson, and M. Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2016, pp. 1–4.

[35] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[36] P. S. Heckbert, "Fundamentals of texture mapping and image warping," Masters' thesis, Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, CA, USA, Rep. UCB/CSD-89-516, Jun. 1989.

[37] J. Jain and A. Jain, "Displacement measurement and its application in interframe image coding," *IEEE Trans. Commun.*, vol. 29, no. 12, pp. 1799–1808, Dec. 1981.

[38] C.-H. Cheung and L.-M. Po, "A novel cross-diamond search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 12, no. 12, pp. 1168–1177, Dec. 2002.

[39] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—including high efficiency video coding (HEVC)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1669–1684, Dec. 2012.

[40] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "HEVC complexity and implementation analysis," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1685–1696, Dec. 2012.

[41] A. Agooun *et al.*, "Acquisition, processing and coding of 3D holoscopic content for immersive video systems," in *Proc. 3DTV-Conf., True Vis.-Capture, Transmiss. Dispaly 3D Video*, Aberdeen U.K., 2013, pp. 1–4.

[42] G. Todor, "Gallery and lightfield data." [Online]. Available: http://www.tgeorgiev.net/Gallery/, Accessed on: Dec. 1, 2016.

Ricardo J. S. Monteiro (M'14) received the Graduate degree in electrical engineering in 2012 and the M.Sc. degree from the Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal, in 2014. He is currently working toward the Ph.D. degree at the University Institute of Lisbon, Lisbon, Portugal.

Since 2010, he has been with the Instituto de Telecomunicações, Portugal, as a Researcher. His current research interests include image and video processing, namely, multiview video and light field image processing and coding.

Paulo J. L. Nunes (M'98) received the Graduate, M.Sc., and Ph.D. degrees in electrical and computers engineering from Instituto Superior Técnico, Universidade Técnica de Lisboa, Lisbon, Portugal, in 1992, 1996, and 2007, respectively.

He is currently an Assistant Professor in the Information Science and Technology Department, University Institute of Lisbon, Portugal, and a member of the Research Staff of Instituto de Telecomunicações, Portugal. His current research interests include 2-D/3-D image and video processing and coding, namely, light field image and video processing and coding.

Dr. Nunes has coordinated and participated in various national and international (EU) funded projects and has acted as a Project Evaluator for the European Commission. He acts often as a Reviewer for various conferences and journals and a member of the program and organizing committees of various international conferences. He has contributed more than 60 papers.

Nuno M. M. Rodrigues (M'99) received the Graduate degree in electrical engineering in 1997, the M.Sc. degree from the Universidade de Coimbra, Coimbra, Portugal, in 2000, and the Ph.D. degree from the Universidade de Coimbra, in 2009, in collaboration with the Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil.

Since 1997, he has been with the Department of Electrical Engineering, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Leiria, Portugal. Since 1998, he has been with the Instituto de Telecomunicações, Portugal, where he is currently a Senior Researcher. He has coordinated and participated as a Researcher in various national and international funded projects. His current research interests include digital signal and image processing, namely, image and video compression, point cloud and light field image and video compression, medical image compression, and many-core programming.

Sérgio M. M. Faria (M'93–SM'08) was born in Portugal in 1965. He received the Engineering degree and the M.Sc. degree, both in electrical engineering, from Universidade de Coimbra, Portugal, in 1988 and 1992, respectively, and the Ph.D. degree in electronics and telecommunications from the University of Essex, Colchester, U.K., in 1996.

Since 1990, he has been a Professor with the Department of Electrical Engineering, Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Portugal. He has collaborated in master courses with the Faculty of Science and Technology and with the Faculty of Economy of Universidade de Coimbra, Portugal. He is an Auditor with A3ES organization for electrical and electronic engineering courses in Portugal. He is a Senior Researcher with Instituto de Telecomunicações.

Dr. Faria's research interests include 2-D/3-D image and video processing and coding, motion representation, and medical imaging. In this field, he has published 1 book, edited 2 books, and authored 7 book chapters, 23 journal papers, and 117 referred conference papers. He has been participating and is responsible for several, national and international (EU), funded projects.

He is an Area Editor of *Signal Processing: Image Communication*. He has been a Scientific and Program Committee Member of many international conferences. He is a Reviewer for several international scientific journals and conferences (IEEE, IET, and EURASIP). In 2014, he received the title of "Agregado" by Instituto Superior Técnico, University of Lisbon.