A 4D DCT-BASED LENSLET LIGHT FIELD CODEC

Murilo B. de Carvalho[1], *Márcio P. Pereira*[2], *Carla L. Pagliari*[3], *Fernando Pereira*[4], *Gustavo Alves*[2], *Vanessa Testoni*[5] *and Eduardo A. B. da Silva*[2]

1. TET/CTC, Universidade Federal Fluminense, Niteroi, Brazil
2. PEE/COPPE/DEL/POLI/UFRJ, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
3. PEE/PGED/IME, Instituto Militar de Engenharia, Rio de Janeiro, Brazil
4. Instituto Superior Técnico, Instituto de Telecomunicações, Lisbon, Portugal
5. Samsung Research Brazil

ABSTRACT

Light fields aim to represent visual information in 3D space. They are 4D structures that contain the images of a given scene from a sampled 2D range of viewpoints. When acquired using a lenslet camera, in addition to the ordinary intra-view redundancy, these views have a great deal of inter-view redundancy. In this work we propose a light field codec that fully exploits the 4D redundancy of light fields by using a 4D transform and hexadeca-trees. It initially divides the light field into 4D blocks and computes a 4D Discrete Cosine Transform of each one. Then the transform coefficients of the 4D block are grouped using hexadeca-trees on a bitplane-by-bitplane basis, and the generated stream is encoded using an adaptive arithmetic coder. The proposed codec has been employed to encode the JPEG Pleno lenslet light fields. The rate-distortion results have been assessed using test conditions comparable to the ones presented at the ICIP 2017 Light Field Coding Grand Challenge. The proposed codec, despite being conceptually simple, achieves competitive rate-distortion performance.

***Index Terms*—** Light fields, image coding, multidimensional transforms

1. INTRODUCTION

Nowadays, with the ever increasing use of visual information in digital form, there is a growing need for more immersive experiences, demanding richer representations of the light in space. A full description of the light rays present in space is given by the 7D plenoptic function [1]. If one simplifies the spectral information by using only 3 color components, assumes that there is no variation in time and considers the intensity of each light ray constant along its path, then one arrives at the so-called light field representation. It is a 4D parameterization of the intensity of a light ray $L(s, t, u, v)$ using the two 2D coordinates of its intersection with two planes: the (s, t) plane (called the view plane) and the (u, v) plane (called the image plane) [2, 3]. If one considers that (u, v) are image coordinates, the variables (s, t) parameterize the direction of the rays passing through (u, v). Alternatively, (s, t) may be considered the viewpoint through which the pixels (u, v) render the scene. In order for one to be able to obtain a high-fidelity reconstruction of a light field from a sampled representation, the sampling densities of (s, t, u, v) must be high enough. Since this representation is 4D, this tends to generate a very large amount of data that is highly correlated, which creates a compelling need for compressed representations.

Thanks to CAPES, CNPq and Samsung Brazil for funding.

Recently, JPEG has issued a call for proposals on light field (LF) coding technologies, namely JPEG Pleno Call for Proposals on Light Field Coding [4–6]. It requested contributions for coding solutions in the area of light fields encompassing, among other things, both lenslet-generated light field images, that are generated with light field cameras [7, 8], and light field images obtained using high density 2D camera arrays (HDCA) [4, 9].

Most recent solutions for light field coding are based on the coding of reference views followed by the prediction/estimation of intermediate views, with or without transmitting a prediction residue [10–15]. These are essentially composed by two types of steps - one type exploits the intra-view redundancy and the other exploits the inter-view redundancy (usually using inter-view prediction). Therefore, in such coding schemes, the joint intra-inter-view redundancy, and thus the overall 4D redundancy of the light fields, tends not to be fully exploited. In this work, we propose a simple coding scheme for the lenslet subaperture images that targets to exploit the inherent 4D redundancy of light fields by employing the classical transformation, quantization and entropy coding three-steps paradigm that has been employed since JPEG-1 times [16]. It does so by using 4D transforms combined with an hexadeca-tree bitplane decomposition followed by entropy coding.

This paper is organized as follows. Section 2 presents previous works on light field image coding, including the ones used as benchmarks to assess the performance of our method, while Section 3 details the proposed coding scheme. In Section 4 the JPEG Pleno datasets and test conditions are presented, followed by the assessment of the rate-distortion performance of the proposed method. Section 5 states the conclusions.

2. RELATED WORK

In this section we briefly overview recent works on light field coding, emphasizing the ones that will be used in the performance assessment in Section 4. The work in [10] and [13] employed HEVC inter coding to compress lenslet light field images arranged as pseudo-temporal sequences. The work proposed in [11] incorporates an inter-view prediction scheme into HEVC inter prediction, exploiting the redundancy in lenslet images using disparity compensation and virtual view reconstruction to compress the lenslet images, while the solution in [12] encodes the light fields using MV-HEVC. The coding schemes presented in [17–19] employ the well known HEVC intra compression efficiency to exploit the redundancies in lenslet images.

In [14] a codec was presented that encodes light fields by partitioning the set of subaperture images of a lenslet dataset into two complementary subsets: set S_A and a dropped set S_B. For example, the set S_A may be composed by the views chosen according to a chessboard-like pattern. The subaperture images in S_A are converted into a pseudo sequence to be compressed with HEVC [20], more specifically the x265 [21] software. Then each subaperture image in S_B is approximated with a weighted sum of the decoded subaperture images in S_A. The prediction coefficients of the set S_B are rearranged as an image that is quantized and lossless compressed. The output coded stream consists of the bitstreams of set S_A and the encoded prediction coefficients. The prediction coefficients and the decoded set S_A are used as inputs to a view synthesis model, producing the decoded set S_B.

The work in [15] presented a lenslet image compression method that is scalable from low bitrates to fully lossless [15]. The lenslet dataset is also partitioned into two sets: the reference subaperture images (views) that are encoded using the JPEG200 standard [22] and a set of dependent views that are reconstructed from the reference views. Their reconstruction is performed employing flexible interpolators implemented by sparse predictors. These are based both on the scene geometry extracted from the depth maps and the geometry of the lenslet array. In addition to the reference views, a region-based segmented version of the depth map is encoded, along to the displacement vectors of each region and the coefficients of the sparse predictors for each region. The solutions in [14] and [15] have also been submitted to the JPEG Pleno Call for Proposals [4]. A transform-based encoding of light fields is presented in [23], performing an empirical rate-distortion analysis of the H.264/AVC and JPEG 2000 Part 10 coding standards applied to integral images.

In the sequel we propose a simple encoder based on the 4D DCT and hexadeca-trees that, unlike the encoders above, aims at exploiting the 4D redundancy of the light field as a whole.

3. THE PROPOSED MULE-TH CODEC

In this section we introduce the proposed codec, the Multidimensional Light field Encoder using 4D Transforms and Hexadeca-trees (MuLE-TH). It exploits the 4D redundancy of the light fields using a 4D transform applied to the light field divided into 4D blocks, followed by a partition of the bitplanes of the generated 4D array of coefficients using hexadeca-trees. The generated bitstream is encoded using an adaptive arithmetic encoder. Its block diagram is illustrated by Figure 1.

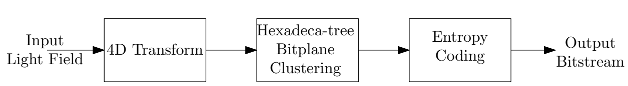

Fig. 1: MuLE-TH block diagram.

The DCT is well know to concentrate the energy of redundant signals in its lower frequency coefficients [24]. The 4D-DCT in MuLE-TH is applied to each light field spatial dimension (separable transform). It thus can take advantage of the redundancy in each dimension to generate coefficients whose energy is well concentrated in the 4D frequency space. The 4D-DCT pipeline is depicted in Figure 2, where (s, t) are the inter-view dimensions and (u, v) are the intra-view dimensions.

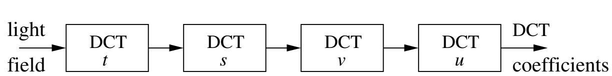

Fig. 2: 4D-DCT pipeline.

Since the DCT generates coefficients that tend to be concentrated in the lower 4D frequencies, after quantization most higher frequency coefficients tend to be zero. Thus an efficient encoding method would be one that can efficiently represent these zeros. When the 2D-DCT is employed, an effective way is to perform zigzag scanning followed by run-length encoding in order to skip the zeros, thus encoding the non-zero coefficient location information using as less symbols as possible [24]. But there are alternative ways to encode the positions of the non-zero coefficients, such as the use of quad-trees [25]. Each node of a quadtree divides a 2-D region into four sub-regions. If a subregion contains only zero coefficients or a single non-zero coefficient, it is not further subdivided and a symbol '0' is encoded. Otherwise, the subregion is further subdivided into four subregions and a symbol '1' is encoded. This subdivision process proceeds recursively until no further subdivision can be performed. At the end of this process, the positions of all the non-zero coefficients are encoded by this '0's and '1's bitstream. The quadtree is the data structure that represents the whole subdivision process. A '0' symbol in a quadtree is thus a way to indicate that all the coefficients inside a region are zero; therefore, if this region is large, this '0' symbol is very efficient in signaling that all those coefficients are zero. Due to this property, quadtrees have been successfully used in image codecs, the one described in [26] being a good example.

Since in MuLE-TH one uses a 4D-DCT, the equivalent to a quadtree would be to perform subdivisions of 4D regions of coefficients into 16 4D subregions. Thus, a '0' indicating no further subdivision of a 4D region is an extremely efficient way to represent many zeros using just one symbol (an $N \times N \times N \times N$ region contains N^4 zeros). One can call such a data structure an hexadeca-tree, where the "hexadeca" prefix stands for division of a 4D region into 16 4D subregions.

In the proposed encoder, after the 4D-DCT is performed, as is done in [26] for the 2D case, the set of coefficients is grouped in bitplanes. A coefficient is considered non-significant on a bitplane if its bits belonging to the bitplanes that are more significant than the current one are all zero. Otherwise, the coefficient is considered as significant. A hexadeca-tree is used to group the non-significant coefficients and thus localize the significant coefficients. The bitplanes are scanned from the most significant to the least significant one, the least significant bitplane used being determined by the desired quantization level. Both the hexadeca-tree bits and the bits from the coefficients are encoded using an adaptive arithmetic coder.

In the sequel we describe the algorithms used to implement each of the steps depicted in the block diagram in Figure 1.

3.1. 4D transform

The 4D-DCT is performed as described in Figure 2. The generated coefficients are represented as 32-bit integers, and grouped into a 4D array of subbands of same frequency transform coefficients.

3.2. Hexadeca-tree bitplane clustering

The bitplanes of the transformed light field at the output of the 4D transform block (Figure 1) are processed by an hexadeca-tree bitplane coder following four steps:

1. *Initialization:* The current bitplane variable, B_c is set to the maximum allowed value, that should be set so that the maximum integer magnitude output by the DCT stage is smaller than 2^{B_c+1}. In the current implementation it is set to 30.
   The variable $B_{\min}$ is set to that $2^{B_{\min}}$ corresponds to the desired quantization step size.

2. *Hexadeca-tree partition of the current bitplane:*

   2.1. The transformed light field is searched and each coefficient is compared to 2^{B_c}.

   2.2. If all coefficient magnitudes are below 2^{B_c}, output a binary segmentation flag '0' and go to Step 3.

   2.3. Otherwise, output a binary segmentation flag '1' and segment the light field in, at most, 16 sub-light fields (a $T \times S \times V \times U$ light field will be segmented into 16 sub-light fields of dimensions $\max\{\frac{T}{2}, 1\} \times \max\{\frac{S}{2}, 1\} \times \max\{\frac{V}{2}, 1\} \times \max\{\frac{U}{2}, 1\}$).[1]

   2.4. Recursively repeat Steps 2.2 and 2.3 for each sub-light field with more than one coefficient.

   2.5. *Transmission of the significant coefficients:* Whenever the dimension of the sub-light fields is $1 \times 1 \times 1 \times 1$, the coefficient value is sent to the entropy encoder, using $\max\{B_c - B_{\min}, 0\}$ bits to encode the magnitude plus one bit to encode the sign (if the magnitude is greater than zero).

   2.6. When no more sub-light fields can be partitioned go to Step 3. Otherwise, repeat Step 2 for each sub-light field.

3. *Current bitplane update:* Decrement B_c and repeat Step 2 for the remaining sub-light fields.

4. *Termination:* The process terminates when $B_c = B_{\min}$ and all coefficients have been encoded.

3.3. Entropy coding

Since the statistics of the segmentation flags and of the DC and AC coefficients are different, the bitstream generated by the hexadeca-tree bitplane clustering is encoded using a context-based binary adaptive arithmetic coder with:

- One binary context for segmentation flags (indicating hexadeca-tree partitions) by bitplane.
- One non-binary context for DC coefficients per bitplane with as many symbols as DC coefficients up to 512 symbols. For more than 512 symbols, the remaining least significant bits are encoded using an equally likely binary model.
- One non-binary context for AC coefficients per bitplane. A similar strategy as for encoding the DC coefficients is used.

4. PERFORMANCE ASSESSMENT

The objective performance assessment is accomplished following the recommendations defined in the JPEG Pleno Call for Proposals [4, 27], which were also used in the ICIP 2017 Grand Challenge on Light Field Image Coding [28]. These will be described in the sequel.

4.1. JPEG Pleno datasets and test conditions

We have employed the lenslet datasets that are listed in the JPEG Pleno Call for Proposals [4] (also in the ICIP 2017 Grand Challenge on Light Field Image Coding [28]). The central subaperture views of the lenslet datasets are displayed in Figure 3. From left to right and top to bottom they are: Bikes, Danger de Mort, Stone Pillars Outside, Fountain&Vincent 2, and Friends 1.

[1] When any dimension $K = T, S, V$, or U is odd, the dimensions of the segments will be given by $\left\lfloor\left(\frac{K}{2}\right)\right\rfloor$ and $K - \left\lfloor\left(\frac{K}{2}\right)\right\rfloor$, respectively.

Fig. 3: Test lenslet light field images (central views).

Table 1: BD-Rate Bjontegaard metric: Bitrate savings comparison using HEVC as reference.

	Bikes	Danger	Pillars	Fountain	Friends
VP9	-22.52	-18.51	-21.71	-18.07	-21.92
USTC	-21.00	NA	NA	NA	NA
TUT	-26.74	-84.52	-51.76	-9.05	-10.06
MuLE-TH	-39.11	-53.19	-38.74	-24.73	-34.83

The luminance and chrominance have been encoded using YUV components with 4:2:2 chroma subsampling, color and gamma corrected for anchor coding (HEVC [20] and VP9 [29] benchmarks), with 10-bit precision and little endian (storage of the least significant byte in the smallest address). Since the boundary views are too dark only the 13×13 central views from the original 15×15 array of views are encoded with the objective evaluation performed on the 13×13 views [27]. The HEVC and VP9 anchors are generated using the HEVC (x.265) [21] and VP9 [29] encoders, both with pseudo-temporal sequences - where lenslet subaperture images are scanned and concatenated following a serpentine scanning order [27]. The PSNR was computed assuming 10-bit dynamic range. According to the JPEG Pleno CfP [27], the proponents should encode the lenslet images at specific target bitrates of: 0.75 bpp, 0.1 bpp, 0.02 bpp and 0.005 bpp.

The mean values of PSNR-YUV, as well as the mean value of SSIM (YUV) are obtained as specified in the JPEG Pleno Call for Proposals [4] ($M_{YUV} = (6M_Y + M_U + M_V)/8$, where M is either PSNR or SSIM).

4.2. Results and analysis

Table 1 summarizes the Bjontegaard bitrate saving comparison (BD-Rate) [30] to the HEVC anchor when the metric used is PSNR-YUV. MuLE-TH refers to the proposed method using blocks of $13 \times 13 \times 15 \times 15$ (13×13 inter-view and 15×15 intra-view) for the luminance and blocks of $13 \times 13 \times 8 \times 8$ for the chrominance. In this paper, the performance benchmarks are taken from the recent ICIP 2017 Light Field Coding Challenge assuming they represent the state-of-the-art. TUT refers to the Tampere University of Technology (TUT) codec [15] and USTC to the University of Science & Technology China (USTC) codec [14] with the set S_A being chosen according to a checkerboard pattern. Note that in [14] only results for the Bikes dataset are provided in comparable conditions. Unfortunately, the results from the Mid-Sweden University [12] (yet complying with the requirements of [28]) could not be included in the comparison because they were obtained assuming a 4:2:0 chroma subsampling for computing rate and PSNR, while the other methods assumed 4:2:2 chroma subsampling.

We note that the BD-rate results for the Mule-TH are superior to the ones of the HEVC and VP9 benchmarks for all lenslet light field images. They are also in general superior to the ones of the method from TUT [15], except for the Pillars and Fountain light field images (the RD performance of the TUT method reported in [15] for these two light field images is far superior to the ones for the other light field images). It is important to note that the TUT light field codec is a quite sophisticated method based on disparity estimation and region-based view synthesis using sparse prediction. Yet, our proposed MuLE-TH codec, based on a simple 4D block DCT achieves superior RD results when compared to the TUT method for 3 of the 5 lenslet lighfield images.

Figures 4–8 show the SSIM-YUV RD performance of MuLE-TH compared to the ones of the anchors HEVC and VP9 (see Subsection 4.1) for the lenslets images. We can also see the superiority of the MuLE-TH performance relative to the ones of the anchors HEVC and VP-9. This is especially relevant for the lower bitrates where gains in quality are typically more difficult to achieve.

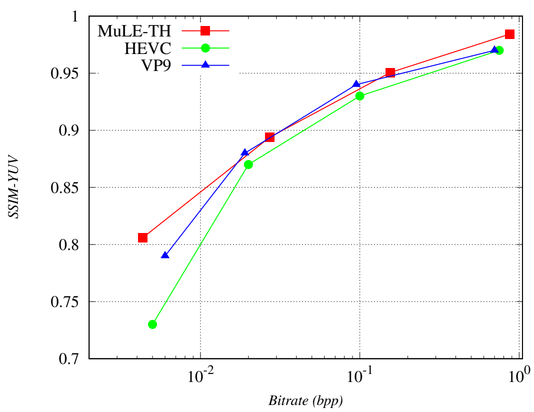

Fig. 4: SSIM RD performances for Bikes.

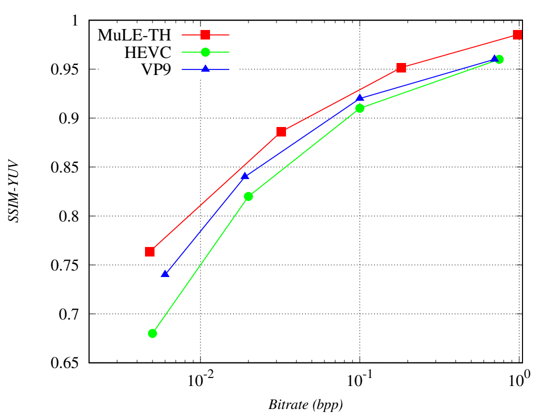

Fig. 5: SSIM RD performances for Danger de Mort.

We can see that the competitive results for MuLE-TH have been achieved using a simple codec, that uses a straightforward block 4D-DCT, hexadeca-tree partitioning of the bitplanes and an adaptive arithmetic coding, that has a great potential of being further refined (e.g., using variable block sizes, prediction, etc.). On the other hand, the two compared ICIP 2017 Grand Challenge methods are based on sophisticated reference view coding together with powerful view synthesis. The anchors use the highly optimized HEVC and VP9 video encoders. This strongly suggests that approaches using coding paradigms that fully exploit the 4D redundancy are worthy investigating.

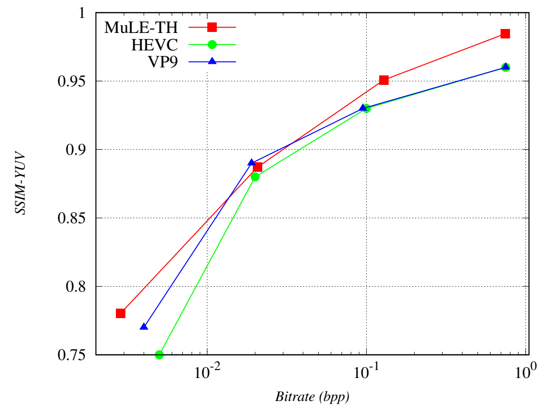

Fig. 6: SSIM RD performances for Stone Pillars Outside.

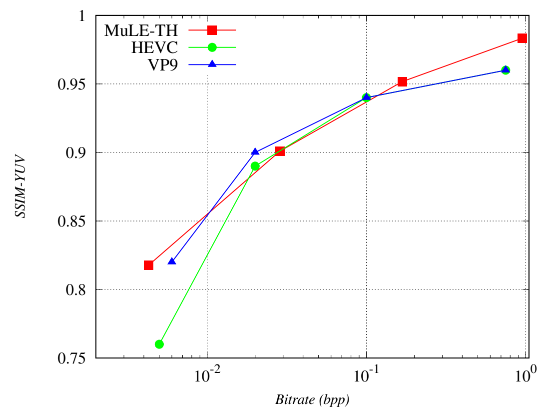

Fig. 7: SSIM RD performances for Fountain&Vincent 2.

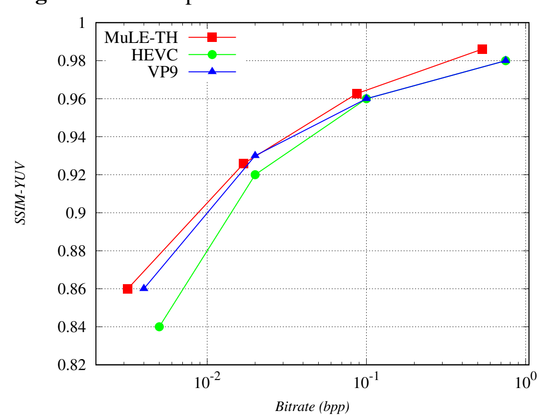

Fig. 8: SSIM RD performances for Friends 1.

5. CONCLUSIONS

In this paper we have proposed a lenslet light field codec based on 4D transforms. The DCT coefficients of the 4D block are clustered using an hexadeca-tree decomposition of their bitplanes. The results for the lenslets JPEG Pleno images are essentially competitive with the ones of the anchors and other methods submitted to the ICIP 2017 Light Field Coding Great Challenge.

It is important to observe that this has been achieved with just a simple encoder that tries to exploit the light fields 4D redundancy as a whole. Therefore, the proposed coding approach, which is naturally able to exploit 4D redundancy, and the results obtained for the lenslet images, suggest that this strategy has a big potential that should be further investigated.

6. REFERENCES

[1] Edward H. Adelson and James R. Bergen, "The plenoptic function and the elements of early vision," *M. Landy and J. A. Movshon, (eds) Computational Models of Visual Processing*, 1991.

[2] Ren Ng, *Digital Light Field Photography*, Ph.D. thesis, Stanford, CA, USA, 2006, AAI3219345.

[3] Donald G. Dansereau and University of Sydney. Australian Centre for Field Robotics, *Plenoptic Signal Processing for Robust Vision in Field Robotics*, University of Sydney, 2014.

[4] ISO/IEC JTC 1/SC29/WG1 N74014, "JPEG Pleno Call for Proposals on Light Field Coding," January 2017.

[5] "Overview of JPEG Pleno," https://jpeg.org/jpegpleno/index.html.

[6] Touradj Ebrahimi, Siegfried Foessel, Fernando Pereira, and Peter Schelkens, "JPEG Pleno: Toward an Efficient Representation of Visual Reality," *IEEE MultiMedia*, vol. 23, no. 4, pp. 14–20, October 2016.

[7] "Lytro," https://www.lytro.com/.

[8] "Raytrix," http://www.raytrix.de/.

[9] "The Stanford Multi-Camera Array," https://graphics.stanford.edu/projects/array/.

[10] Cristian Perra and Pedro Assunção, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[11] Yun Li, Marten Sjöström, Roger Olsson, and Ulf Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Transactions on Image Processing*, vol. 25, no. 1, pp. 80–91, January 2016.

[12] Waqas Ahmad, Roger Olsson, Marten Sjöström, "Interpreting Plenoptic Images as Multi-View Sequences for Improved Compression," in *IEEE International Conference on Image Processing 2017 - Light Field Coding Grand Challenge*, Beijing–China, September 2017.

[13] Dong Liu, Lizhi Wang, Li Li, Zhiwei Xiong, Feng Wu, and Wenjun Zeng, "Pseudo-sequence-based light field image compression," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[14] Shengyang Zhao, Zhibo Chen, "Light field image coding via linear approximation prior," in *IEEE International Conference on Image Processing 2017 - Light Field Coding Grand Challenge*, Beijing–China, September 2017.

[15] Ioan Tabus, Petri Helin, Pekka Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000," in *IEEE International Conference on Image Processing 2017 - Light Field Coding Grand Challenge*, Beijing–China, September 2017.

[16] William. B. Pennebaker and Joan L. Mitchell, *JPEG Still Image Data Compression Standard*, Van Nostrand Reinhold, 1993.

[17] Yun Li, Roger Olsson, and Marten Sjöström, "Compression of unfocused plenoptic images using a displacement intra prediction," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[18] Caroline Conti, Paulo Nunes, and Luis D. Soares, "HEVC-based light field image coding with bi-predicted self-similarity compensation," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[19] Ricardo Monteiro, Luis Lucas, Caroline Conti, Paulo Nunes, Nuno Rodrigues, Sergio Faria, Carla Pagliari, Eduardo A. B. da Silva, and Luis D. Soares, "Light field HEVC-based image coding using locally linear embedding and self-similarity compensated prediction," in *2016 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, July 2016, pp. 1–4.

[20] *ITU-T & ISO/IEC.*, "High Efficiency Video Coding, Rec. ITU-T H.265 and ISO/IEC 23008-2," 2013.

[21] "x.265," http://x265.org/.

[22] "ISO/IEC 15444-1:2016 Preview Information technology – JPEG 2000 image coding system," .

[23] R. Olsson, "Empirical Rate-Distortion Analysis of JPEG 2000 3D and H. 264/AVC Coded Integral Imaging Based 3D-Images," in *2008 3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, May 2008, pp. 113–116.

[24] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann, 4th edition, 2012.

[25] Hanan Samet, "The quadtree and related hierarchical data structures," *ACM Comput. Surv.*, vol. 16, no. 2, pp. 187–260, June 1984.

[26] Jim Andrew, "A simple and efficient hierarchical image coder," in *Proceedings of International Conference on Image Processing*, October 1997, vol. 3, pp. 658–661 vol.3.

[27] ISO/IEC JTC 1/SC29/WG1N75024, "JPEG Pleno Call for Proposals Submission Process Details," March 2017.

[28] "ICIP 2017 Light Field Image Coding Grand Challenge," http://2017.ieeeicip.org/GC1.aspl.

[29] "VP9 Video Codec," https://www.webmproject.org/vp9/.

[30] Serge Matyunin, "Bjontegaard metric calculation (BD-PSNR) - version 1.0 from MATLAB Central File Exchange," 2013.