Microlens Image Sparse Modelling for Lossless Compression of Plenoptic Camera Sensor Images

Ioan Tabus and Petri Helin
Tampere University of Technology
Laboratory of Signal Processing
P.O. Box 553, FI-33101, Tampere, Finland

***Abstract*—This paper studies the lossless compressibility of raw sensor images acquired by plenoptic cameras, when optimally interpolating the microlens images in terms of already encoded microlens images. The geometrical information necessary for splitting the sensor image into projections of microlenses, together with a relatively small bitstream for encoding the raw image at the microlens centers are encoded as a first stage. The scanning order for sampling the data from the sensor follows row-by-row the approximate hexagonal lattice pattern of the microlenses, and the pixels inside each microlens are scanned in an ascending spiral order. The predictive encoding of a pixel from a microlens block uses the similarly located pixels (possibly slightly shifted) in the blocks from nine closest causal microlenses (those already encoded) and the pixels from its own microlens located in the encoded part of the spiral. A minimum description length optimal sparse predictor is designed for each microlens. The sparsity masks and prediction coefficients are encoded in a second stage and the prediction errors at every pixel are finally encoded in a third stage, in a view-by-view order (a view index being determined by the pixel's index in its block), using contexts accounting for the magnitude of errors at views already encoded. The experimental results show better performance than the JPEG 2000 image standard applied on the raw image.**

I. Introduction

A plenoptic camera interposes between the sensor and the optical lens system of a traditional camera an $m_r \times m_c$-array of microlenses (in the order of hundreds of thousands microlenses), that will record at certain pixels in the sensor the light coming from specific directions. Encoding the raw image obtained at the sensor is studied in this paper. Lossless compression of raw sensor images from contemporary plenoptic cameras was discussed earlier in [1], where the sensor image is split into four subimages, one for each Bayer phase, then repetitive rectangular block structures are searched in the image, and the differences between matching blocks are encoded using Lempel-Ziv coding. We present here a more involved compression scheme, exploiting the known arrangement of microlenses in a hexagonal lattice, performing sparse prediction for designing the best interpolation between the approximate hexagonal microlens images (MI) to obtain a prediction for the MI to be encoded, and encoding the errors using context encoding.

The raw image is obtained by recording the light coming from different directions, at the pixels under each microlens. We deal with encoding the raw image, which is an $n_r \times n_c$ array of scalar values represented on 10 bits, where demosaicing was not applied yet. From this primary image one can obtain by demosaicing and devignetting the RGB lenslet image and by further processing the LF structure of subaperture images. The Bayer filter of the camera produces a checkerboard pattern, that is visible even for the white areas of the scene. Each microlens is visible in the raw image as a circular pattern with a diameter of about 15 pixels. For example in Fig. 1 (a)-(c) we show one raw image at several zoom levels. ‘

We take advantage of the prior knowledge of the geometry of microlens arrangement, so that the repetitive patterns are well defined a priori and no expensive block matching process needs to be carried on. The basic geometric fact to be used is that the microlens centers are located at the nodes of a hexagonal lattice, defined in the coordinate system of the sensor. The technological reason is that the hexagonal lattice of disk centers offers the best coverage of the sensor by disks of a given radius.

The hexagonal lattice arrangement of the microlenses centers is illustrated in Fig. 1 (c)-(d), where the Voronoi cells determined by the centers are superimposed over the sensor image, showing a very good match with the circular patterns readily identifiable on the sensor image. We refer in the following loosely to each hexagonal Voronoi cell (or to the circularly shaped set of pixels inscribed in it) in Fig. 1 (c) as a microlens image (MI), but more precise definitions will be introduced later. The sensor image is hence decomposed into a collection of MIs. The main theme of this paper is designing optimal interpolators for the pixels of one MI as functions of the pixels in other MIs.

The paper is organized as follows: Section II introduces first the data structures for sampling the light field from the raw image. Then the prediction problem for interpolating one MI from neighbour MIs is presented and shown to be solved as a sparse regression modelling problem. In Section III the encoding and decoding algorithms are presented and Section IV presents experimental results, with Section V drawing conclusions.

II. Sampling the sensor image into a plenoptic lightfield data structure

In this section we exemplify the numerical values of parameters with those taken from one raw LFR data structure of a Lytro Illum camera, corresponding to the *Bikes* test image from the dataset used in the experimental section.

ISBN 978-0-9928626-7-1 © EURASIP 2017

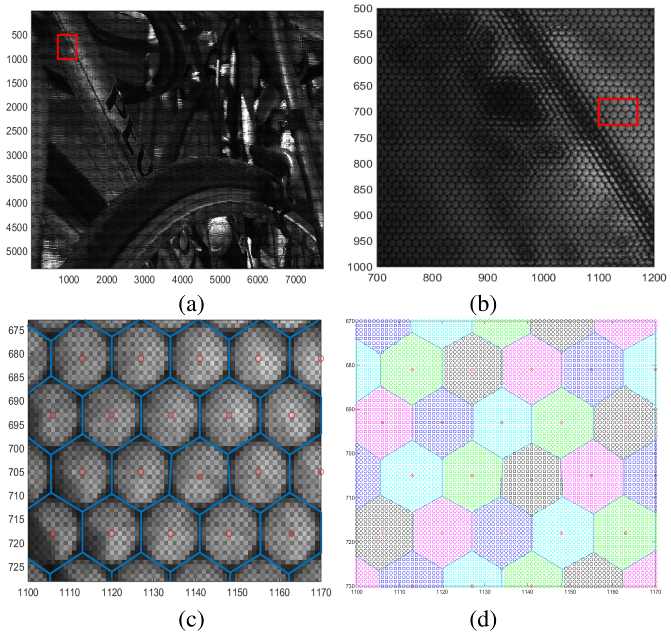


Fig. 1. (a) Raw sensor image "Bikes" having at each pixel a graylevel value represented on 10 bits. (b) Zoom in the rectangular area marked in a), where the circular patterns (of diameter about 15 pixels) are the projections of the microlenses on the sensor; (c) Zoom in the rectangular area marked in panel b), where the hexagonal grid of microlense centers is estimated with the parameters $\mathbf{\Psi}$ provided by the lightfield processing toolbox [2]; the centers have integer coordinates marked by red circles; the Voronoi cells determined by the lattice of centers are drawn with blue lines; (d) The pixels marked with same color are alocated to same Voronoi cell. The pixels equally close of two MI-centers are allocated to the MI appearing first in the scanning order. The smallest number of pixels of a microlens Voronoi cell is 168, the largest is 189, and the most frequent (in almost half of the cases) is 175.

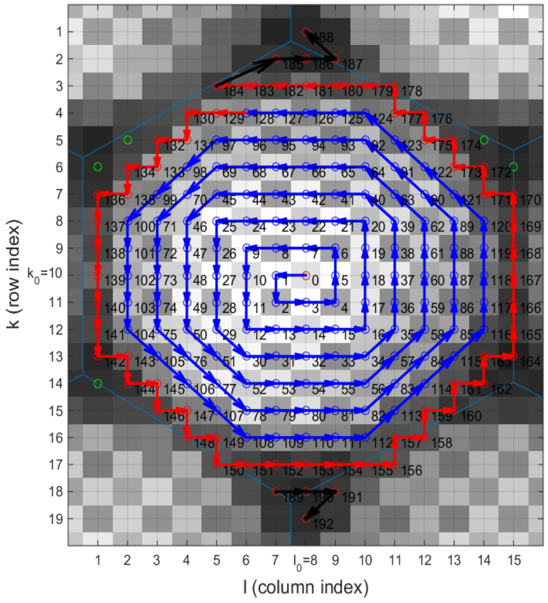


Fig. 2. Spiral scanning of a microlens block. The template $\mathbf{T}$ alocates an index $T(k,l) \in \{0,\dots 192\}$ to the pixel at (k,l), e.g. $T(10,8) = 0$ and $T(10,7) = 1$. The template selects 193 pixels: the first 129 pixels, reached by the blue arrows in the spiral, are contained in a circle of radius 6.4 centered at $(0,0)$; the next 56 pixels, reached by the arrows marked in red, have the centers within a circle of radius 7.7; the last 6 pixels in the template, marked from 185 to 192 and reached by black arrows, are normally outside the area covered by the microlense, and are added only to cover the interstitia between the circular microlenses. The elements of each microlens block (i,j) selected by the template are forming the 192-dimensional vector $\mathbf{v}^{(i,j)}$.

The $(n_r \times n_c) = (5368 \times 7728)$ sensor image $\mathbf{Y}$, having elements denoted $Y(x,y)$, records the light coming through an array of microlenses. The centers of the microlenses are arranged in a $(m_r \times m_c) = (434 \times 541)$ hexagonal grid, where one row contains m_c centers equally spaced at $L_m \approx 14.29$ pixels apart, and the m_r rows are equally spaced vertically at distance $\frac{\sqrt{3}}{2}L_m \approx 12.37$. In average one MI should be allocated 176.7 pixels.

Each node in the ideal hexagonal grid of centers has real valued center coordinates $(x'_C(i,j), y'_C(i,j))$, that are fully determined by the parameters of the hexagonal grid, $\mathbf{\Psi}_m = (L_m, h_o, v_o, \alpha_m)$, estimated using the Light Field Toolbox[1], from the calibration files provided by the camera manufacturer together with each camera. The parameters include a rotation angle α_m and the horizontal h_o and vertical offset v_o of the hexagonal grid with respect to sensors coordinate system.

These parameters are needed along with the raw image for obtaining the rectified lightfield image, by the involved processing described in [2]. No usage of the raw data is possible without the knowledge of $\mathbf{\Psi}_m$, hence these parameters are always known and are transmitted to the decoder. In this paper we use instead of ideal real valued centers $(x'_C(i,j), y'_C(i,j))$ their values rounded to integers, denoted $(x_C(i,j), y_C(i,j))$

We define a light field structure, $L_Y(k,l,i,j)$ where (k,l) are indices of a subaperture image (a.k.a view), and (i,j) are the pixel indices in the view. The lightfield structure organizes the pixels from the raw sensor image, by covering the sensor image with the microlenses, where the integer centers of the microlens(i,j) in the coordinate grid of the sensor are $(x_C(i,j), y_C(i,j))$. The sampling equation for defining the lightfield structure is

$$L_Y(k,l,i,j) \quad = \quad Y(x_C(i,j)+k-k_0, y_C(i,j)+l-l_0).$$

The goal is covering the sensor image by disks of pixels with centers at $(x_C(i,j), y_C(i,j))$. We note that on a discrete grid a "circularly" shaped set of pixels is well approximated by the regular octagon which includes all 129 pixels along blue arrows in Fig. 2, but that disk template will leave uncovered a large proportion of pixels from the sensor. The next largest disk is a regular octagonal structure of pixels that includes also the pixels marked by red arrows, containing 185 pixels. Two such large disks, when placed at the neighbor centers, will sometimes overlap several pixels from their outer borders. Since we need to represent each pixel in the sensor image as belonging to one disk, we decide to choose the 185 pixels disk as a basis for the sampling template.

We introduce the sampling template illustrated in Fig. 2, where the used pixels are marked in the rectangle $k = 1,\dots,19$, $l = 1,\dots,15$ with labels from 0 to $K = 192$. The center of the template is at $k_0 = 10$ and $l_0 = 8$. The main part of the template is formed within the 15×15 square from rows 3 to 17 and contains $K_1 = 185$ pixels; the additional pixels labeled 185 to 192 in rows $1, 2, 18, 19$ are covering

[1] https://se.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4

ISBN 978-0-9928626-7-1 © EURASIP 2017

areas outside the circles associated to microlenses, but which anyway must be encoded to have a complete representation of the raw sensor image for lossless compression. The index in the template will be identified with the index of a direction of view, and plays an important role in the conditional coding part of our encoding algorithm.

The chosen template covers sufficiently well the hexagonal Voronoi cells, which are of sizes varying from 164 to 189 pixels (see Fig. 1 (d)). To illustrate the covering realised by the template, in the Fig. 2 it was shown as a background the worst case, of a Voronoi cell drawn in blue lines having the largest number of pixels, 189. In this worst case, a few pixels, with centers marked in green, are not covered by the template. These non-covered pixels, together with some pixels at the border of the sensor are encoded in an initial phase (Stage I of Algorithm 1).

The 185 pixels at the pairs (k,l) within the wide circle of radius $k^2+l^2 \leq 7.7^2$ are pixels with reliable measurements, since they are the pixels under the circular shape of the microlens, and are receiving much more light than the pixels in the corners of the 15×15 square. Since not all pixels in the rectangle addressed by (k,l) are associated with the microlens (i,j), the spiral indexing function $T(k,l) \in 0,\ldots,192$ will be used to sample from $\mathbf{Y}$ the block of pixels belonging to microlens (i,j), under the form of a 193-dimensional vector $\mathbf{v}^{(i,j)}$ with the elements indexed by $\tau = T(k,l)$,

$$\begin{aligned}\mathbf{v}^{(i,j)}_{T(k,l)} &= L_Y(k,l,i,j) \\ &= Y(x_C(i,j)+k-k_0, y_C(i,j)+l-l_0), \quad (1)\end{aligned}$$

where the first element, $\mathbf{v}^{(i,j)}_0$, represents the value at the central pixel of microlens (i,j) and the next elements, $\mathbf{v}^{(i,j)}_1, \mathbf{v}^{(i,j)}_2, \ldots$ are advancing along the spiral. Denoting the template subindices (k_τ, l_τ), in terms of the template index $T(k_\tau, l_\tau) = \tau$ we can write directly the spiral elements as $\mathbf{v}^{(i,j)}_\tau = Y(x_C(i,j)+k_\tau-k_0, y_C(i,j)+l_\tau-l_0)$.

The first elements $\mathbf{v}^{(i,j)}_0$ in all microlenses are encoded in the first stage as a side information, and what remains to be encoded at each microlens is the $K=192$-dimensional vector $[\mathbf{v}^{(i,j)}_1 \;\ldots\; \mathbf{v}^{(i,j)}_K]^\top$ which is denoted for brevity $\mathbf{v}^{(i,j)}_{1:K}$, while the full vector is denoted $\mathbf{v}^{(i,j)}_{0:K}$.

For realizing flexible interpolation functions one needs to sample the sensor image using the spiral template centered at arbitrary coordinates (x,y) in the sensor image; hence we introduce for all $\tau = 0,\ldots,192$ the sampling function:

$$\mathbf{w}^{(x,y)}_\tau = Y(x+k_\tau-k_0, y+l_\tau-l_0), \quad (2)$$

which constructs the 193-dimensional vector $\mathbf{w}^{(x,y)}$ corresponding to the spiral located at any coordinates (x,y) in the raw sensor image. For all $\tau = 0,\ldots,192$ this vector will be identical to the vector (1) at the center of the microlens (i,j):

$$\begin{aligned}&\mathbf{w}^{(x_C(i,j),y_C(i,j))}_\tau \\ &= Y(x_C(i,j)+k_\tau-k_0, y_C(i,j)+l_\tau-l_0) = \mathbf{v}^{(i,j)}_\tau.\end{aligned}$$

Now we can formulate the regression problem as predicting the vector $\mathbf{v}^{(i,j)}_{1:192}$ (the elements labeled $1,\ldots,192$ from the spiral at the microlens (i,j)) as a linear combination of the following vectors: 1) the all-1 column vector $\mathbf{1}$ of length K, accounting for a possible bias term; 2) the vector containing the elements labeled $0,\ldots,191$ of $\mathbf{v}^{(i,j)}$, which ensures predicting the τth element of $\mathbf{v}^{(i,j)}$ by its $(\tau-1)$th element, i.e. each element in the current spiral is predicted by the value of its preceding element in the spiral; 3) finally the vectors $\mathbf{w}^{(x_C(i',j')+x_0,y_C(i',j'))+y_0)}_{1:192}$ realizing cross-prediction based at the spirals at the neighbor microlenses (i',j') and the spatially shifted versions of those spirals by maximum shifts of 1 in all cardinal directions. The prediction can be written formally as the linear combination

$$\begin{aligned}&\hat{\mathbf{v}}^{(i,j)}_{1:192} = \psi_0 \cdot \mathbf{1} + \psi_1 \cdot \mathbf{v}^{(i,j)}_{0:191} \quad (3)\\ &+ \sum_{\substack{(i',j')\,\in\,\mathcal{N}_M(i,j)\\ (x_0,y_0)\,\in\,\mathcal{N}_S}} \theta_{(i-i',j-j',x_0,y_0)} \cdot \mathbf{w}^{(x_C(i',j')+x_0,y_C(i',j'))+y_0)}_{1:192}\end{aligned}$$

where the neighbor set $\mathcal{N}_M(i,j)$ selects the nine causal neighbor microlenses, as shown in Fig. 3 (b). Due to the hexagonal grid properties, odd and even rows have their indices shifted by half L_m, so that neighbors in the lattice get different indices depending on the parity of j: for odd j the set is $\mathcal{N}_M(i,j) = \{(i,j-2),(i,j-1),(i-1,j-2),(i-1,j-1),(i-1,j),(i-1,j+1),(i-2,j-1),(i-2,j),(i-2,j+1)\}$, while for even j the set is almost the same, with the change that the elements in the row $i-1$ are replaced by $(i-1,j-1),(i-1,j),(i-1,j+1),(i-1,j+2)$. The set of spatial displacements (x_0,y_0) contains nine pairs, $\mathcal{N}_S = \{(x_0,y_0)|x_0=-1,0,1; y_0=-1,0,1\}$.

The prediction residuals

$$\boldsymbol{\varepsilon} = \mathbf{v}^{(i,j)}_{1:192} - \hat{\mathbf{v}}^{(i,j)}_{1:192} \quad (4)$$

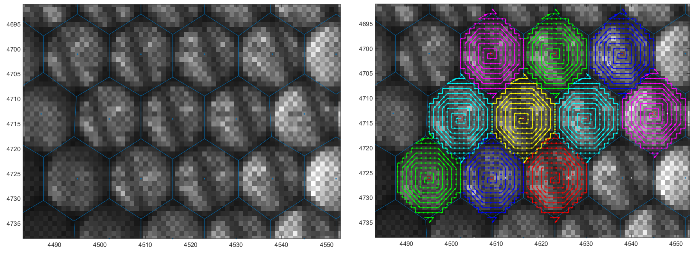
are computed for all microlenses and then encoded and transmitted to the decoder, as presented in the next section.

At the decoder the prediction in (3) is performed element by element along the spiral, and once the τth prediction $\hat{\mathbf{v}}_\tau$ was computed, the element $\mathbf{v}_\tau$ can be perfectly reconstructed using the τth residual ε_τ, hence $\mathbf{v}_\tau$ can be used in the prediction of next element, $\hat{\mathbf{v}}_{\tau+1}$.

The vectors in the right hand side of (3), i.e., the vector $\mathbf{1}$, $\mathbf{v}^{(i,j)}_{0:191}$ and all 81 vectors $\mathbf{w}^{(x_C(i',j')+x_0,y_C(i',j'))+y_0)}_{1:192}$ can be arranged as columns in the regression matrix $\mathbf{A}$. Correspondingly, the parameter vector $\boldsymbol{\Theta}$ in the regression problems has ψ_0 and ψ_1 as the first two elements, and the rest of 81 elements are all coefficients $\theta_{(i-i',j-j',x_0,y_0)}$ that appear in (3). For the microlens (i,j), at the encoder the true values $\mathbf{d} = \mathbf{v}^{((i,j))}_{1:192}$ are known, and one needs to find the parameters $\boldsymbol{\Theta}$ so that

$$\mathbf{d} = \mathbf{A}\boldsymbol{\Theta} + \boldsymbol{\varepsilon} \quad (5)$$

with the optimality criterion being the overall codelength needed for encoding the residuals $\boldsymbol{\varepsilon}$ and the parameters $\boldsymbol{\Theta}$, that needs to be minimized. Since the number of unknowns in (5) is rather small, here there are 192 unknowns in the spiral, and the number of unknown parameters in $\boldsymbol{\Theta}$ is high, in here 83 parameters, the problem of selecting the relevant regressors is



(a) (b)

Fig. 3. Example of block prediction. The block marked with the red spiral in panel b) has to be encoded, conditional on its nine closest causal neighbor blocks (marked with spirals of various colors). The original blocks are shown in panel a), the block to be encoded being quite similar to several of the regressor blocks. Each neighbor block is included in the regression matrix in nine copies, with displacements of maximum one in each axis and direction.

essential. Hence we consider solving (5) as a sparse regression problem where only a small number of regression coefficients are non-zero, and we solve the sparse regression using the minimum description length sparse prediction approach used in [3]. Our sparse predictor implementation is very similar to [3], except that instead of using OMP in the iterative loop for choosing the regression elements we use fast order recursive least squares as in [4]. The optimal sparse solution $(\boldsymbol{\gamma}, \boldsymbol{\theta})$ is defined by the binary selector vector $\boldsymbol{\gamma}$, also called sparsity mask, which selects from the full regressor vector $\boldsymbol{\Theta}$ the nonzero coefficients, $\boldsymbol{\theta} = \boldsymbol{\Theta}_{\boldsymbol{\gamma}}$.

The previous presentation assumed that all the 192 elements in the vector $\mathbf{v}_{1:192}^{((i,j))}$ are unknown and need to be encoded. Due to the asymmetries of the Voronoi cells, some of the elements of $\mathbf{v}_{1:192}^{((i-1,j))}$ and $\mathbf{v}_{1:192}^{((i,j))}$ have some pixels in common, which are encoded as part of MI $(i-1,j)$ and need not be encoded in MI (i,j). This is tackled very easily, when processing MI (i,j), simply by removing from $\mathbf{d}$ and $\mathbf{A}$ in the regression system (5), the rows corresponding to the already encoded pixels. Hence, the number of equations in the sparse regression system varies from one MI to another.

III. The encoding and decoding algorithms

First the encoder writes the parameters of the hexagonal lattice $\boldsymbol{\Psi}$ (six floating point numbers) into the bitstream, and hence the decoder can identically reconstruct the partition of pixels into MIs using the template function.

Then the encoder encodes the raw image values $Y(x_C(i,j), y_C(i,j))$ at all center pixels, e.g., by JPEG 2000. The reason for transmitting the centers in the first pass is to be able to perform prediction within the microlens spiral, starting from the known value of Y at the microlens center and then advancing in spiral order and predicting the element τ based on the element $\tau - 1$. This is implemented by the second term in the predictor (3).

The second stage transmits the value of Y at the irregular pixels, where we cannot apply sparse predictive coding, due to missing conditioning information, and also the values of pixels non-covered by a template (e.g the green pixels in Fig. 2). The microlenses located at the sensor borders (rows $i = 1, 2, 3, m_r - 2, m_r - 1, m_r$ and columns $j = 1, 2, 3, m_c - 2, m_c - 1, m_c$) are checked if they have enough pixels at which to apply the full prediction model (3) or a partial model using only one MI neighbor. If in the current microlens there are not more than 60 pixels for which the model can be applied, all microlens pixels are transmitted in the first pass, without sparse prediction. The sequence of first pass pixels is denoted $\mathcal{F} = z_{F_1}, \ldots, z_{F_{n_F}}$, and their locations are identically found by the decoder, by analyzing the geometry of the hexagonal grid. The encoder transmits then to decoder the values in the sequence $\mathcal{F}$ differentially, $z_{F_i} - z_{F_{i-1}}$ using arithmetic coding, resulting in $\mathcal{L}(\mathcal{F})$ bits. The decoder at this time has information about $\mathbf{Y}$ at the centers of the microlenses and at the location of irregular pixels.

The third stage is the regular sparse predictive modelling, which is done in a line-by-line scanning order of the microlenses, as presented in the previous section. In this third stage the sparsity masks and prediction coefficients for all microlenses are encoded.

The fourth stage is encoding of the prediction errors (4), which are seen as an $(n_r \times n_c)$ image $\mathbf{R}$ (residual image) that needs to be transmitted independently of the previous prediction stage. The same sampling function used for $\mathbf{Y}$ is used to obtain a lightfield data structure, denoted $\{L_R(k,l,i,j)\}$, corresponding to $\mathbf{R}$. The τ-th image, $L_R(k_\tau, l_\tau, \cdot, \cdot)$, in the data structure is encoded conditional on the context image $\mathbf{E}_{\tau-1}$, defined as the average of log-absolute values of elements in $L_R(k_{\tau-1}, l_{\tau-1}, \cdot, \cdot)$. For a robust average evaluation, a median operator is used over the defined elements in a 5×5 windows.

The pseudocode of the algorithm is shown as Algorithm 1.

A. Decoding the image

The decoding algorithm performs similar operations as described for encoding (reading the Algorithm 1 and replacing "encode" by "decode"). The only major change is that Stage IV is performed at the decoder before Stage III, since the image of prediction errors $\mathbf{R}$ needs to be the first decoded, which is possible, not needing elements of the image $\mathbf{Y}$. Then, the decoder will reconstruct the image $\mathbf{Y}$, by scanning in the same way the sensor image as in the Stage III of the encoder and using the decoded predictor parameters and the already reconstructed values from $\mathbf{Y}$ (reconstructed using all already known residuals form $\mathbf{R}$) for performing the prediction.

One important implementation detail is that a $(n_r \times n_c)$-binary image $\mathbf{I}$, marking the available pixels when decoding the current MI, is kept at both encoder and decoder, being updated as soon as a pixel of $\mathbf{Y}$ can be reconstructed. When constructing the regression matrix $\mathbf{A}$ for a microlens image, we check at both encoder and decoder, through the marking in $\mathbf{I}$, the availability of the elements to be included in $\mathbf{A}$. If at least one element in a column of $\mathbf{A}$ is not available, that column is removed from the matrix $\mathbf{A}$. This solves conflicting situations, e.g., when the spiral from MI at the left of the current MI is shifted one pixel to the right, it may include one or more elements from the current MI, which are not available.

Algorithm 1 Encoder

```
1: Read the $(n_r \times n_c)$ raw sensor image to be encoded, $\mathbf{Y}$,
   with $Y(i,j) \in 0,\dots,2^{10}-1$.
2: Stage I. Estimate from the *LFP* structure the parameters
   $\mathbf{\Psi}$ of the $(m_r \times m_c)$ hexagonal grid of microlens centers
   $\mathcal{C} = \{[x_C(i,j) \;\; y_C(i,j)], 1 \le i \le m_r, 1 \le j \le m_c\}$ [2].
     I.1 Encode the parameters $\mathbf{\Psi}$.
     I.2 Encode by JPEG 2000 the raw image values
     $Y(x_C(i,j), y_C(i,j))$ at all center pixels from $\mathcal{C}$.
     Total codelength at Stage I: $\mathcal{L}(\mathcal{C})$.
3: Stage II. Go over the hexagonal grid of microlens centers
   in row-wise order, and determine the sequence of first pass
   pixels, $\mathcal{F} z_{F_1},\dots,z_{F_{n_F}}$, which are the pixels that are not
   reached by any template overlaid at the centers from $\mathcal{C}$,
   or are within the irregular MI at the border of the image.
     II.1 Encode the pixels in $\mathcal{F}$ differentially, $z_{F_i} - z_{F_{i-1}}$
     using arithmetic coding, resulting in $\mathcal{L}(\mathcal{F})$ bits.
4: Stage III. Go over the hexagonal grid of microlens centers
   from $\mathcal{C}$ in row-wise order, and process the current
   microlens block, MI (i,j).
     III.1 Formulate the regression problem, by selecting the
     possible regressors as in (3).
     III.2 Solve the sparse regression problem (5). Encode the
     parameters of the sparse predictor at the current MI (i,j)
     (sparsity mask $\boldsymbol{\gamma}$, its size, $|\boldsymbol{\gamma}| = \sum_i \gamma_i$, and non-zero
     coefficient values $\boldsymbol{\theta} = \mathbf{\Theta}_{\boldsymbol{\gamma}}$).
     III.3 Compute the predictions for the current block with
     its optimal sparse predictor parameters. The prediction
     residual for pixel $Y(x,y)$ in the block is stored in the
     residual image $R(x,y)$.
   Total codelength at Stage III: $\mathcal{L}(\boldsymbol{\gamma}) + \mathcal{L}(|\boldsymbol{\gamma}|) + \mathcal{L}(\boldsymbol{\theta})$
5: Stage IV. Resample the image of residuals $\mathbf{R}$ into a
   lightfield structure $L_R(k_\tau, l_\tau, i, j)$, $\forall \tau = 1,\dots,192$.
     IV.1 For $\tau \in \{1,\dots,192\}$
           For all $(i,j) \in \mathcal{C}$ encode $L_R(k_\tau, l_\tau, i, j)$ using
     $E_{\tau-1}(i,j)$ as a context (initializing $E_0(i,j) = 1$).
     IV.2 Compute the $(m_r \times m_c)$ context image $\mathbf{E}_\tau$ as the
     average of log-absolute-errors, using a $5 \times 5-$median
     filter on the image $\log_2 |L_R(k_\tau, l_\tau, :, :) + 1|$.
6: Overall codelength of encoded file is
   $\mathcal{L}_{SMC} = \mathcal{L}(\mathcal{C}) + \mathcal{L}(\mathcal{F}) + \mathcal{L}(|\boldsymbol{\gamma}|) + \mathcal{L}(\boldsymbol{\gamma}) + \mathcal{L}(\boldsymbol{\theta}) + \mathcal{L}(\mathbf{R})$
```

IV. Results

We use data from Light-Field Image Dataset[2], that contains raw data captured with a Lytro Illum camera and select the same subset of twelve scenes as in the recent challenge at ICME 2016. Table I presents the breakdown of the overall codelength on the various partial codelengths for each encoding stage, showing the very small part of the model cost in the overall bit budget (including the cost of sparsity pattern $\boldsymbol{\gamma}$, of its size $|\boldsymbol{\gamma}|$ and of prediction coefficients $\boldsymbol{\theta}$).

	Preliminary Info		Sparse predictors			Resid.	Total
Im.	$\mathcal{L}(\mathcal{C})$	$\mathcal{L}(\mathcal{F})$	$\mathcal{L}(\|\boldsymbol{\gamma}\|)$	$\mathcal{L}(\boldsymbol{\gamma})$	$\mathcal{L}(\boldsymbol{\theta})$	$\mathcal{L}(\mathbf{R})$	$\mathcal{L}_{SMC}$
1	0.042	0.056	0.013	0.104	0.172	5.378	5.764
2	0.041	0.046	0.013	0.101	0.166	5.199	5.566
3	0.042	0.051	0.013	0.099	0.164	5.423	5.793
4	0.041	0.050	0.013	0.099	0.162	5.476	5.840
5	0.037	0.045	0.012	0.099	0.160	4.816	5.169
6	0.040	0.060	0.014	0.107	0.177	5.310	5.708
7	0.045	0.065	0.012	0.097	0.158	6.170	6.547
8	0.041	0.059	0.014	0.107	0.178	5.323	5.722
9	0.042	0.055	0.014	0.106	0.178	5.543	5.939
10	0.041	0.064	0.013	0.103	0.170	5.371	5.762
11	0.040	0.063	0.013	0.102	0.168	4.741	5.127
12	0.043	0.050	0.013	0.107	0.177	5.396	5.786

TABLE I
BREAKING DOWN THE TOTAL CODELENGTH IN BPP OF THE PROPOSED METHOD (SMC) INTO INDIVIDUAL ELEMENTS

Image	$\mathcal{L}_{SMC}$	$\mathcal{L}_{JP2}$	$\Delta = \mathcal{L}_{JP2} - \mathcal{L}_{SMC}$	Δ %
1	5.76	6.24	0.48	7.62
2	5.57	5.74	0.17	3.04
3	5.79	5.93	0.14	2.32
4	5.84	6.02	0.18	2.99
5	5.17	5.70	0.53	9.31
6	5.71	6.69	0.98	14.69
7	6.55	7.24	0.69	9.57
8	5.72	6.79	1.07	15.72
9	5.94	6.49	0.55	8.48
10	5.76	6.23	0.47	7.51
11	5.13	6.08	0.95	15.68
12	5.79	6.67	0.88	13.25

TABLE II
COMPARING THE TOTAL CODELENGTH IN BPP OF THE PROPOSED METHOD AND THE JPEG 2000 METHOD

Using the standard JPEG 2000 in the form implemented in Matlab function imwrite/imread one gets the column marked as JP2 in Table II, where the results of the proposed method, denoted here sparse modelling compression *SMC* are also presented, showing improvements over all files with respect to the standard JPEG 2000. All encoded files were checked for lossless decoding of the original.

V. Conclusions

We studied here the compressibility of the raw image, and we showed that it is compressible to about 5-6 bpp, much better than the uncompressed representation using 10 bits. The proposed method surpasses the performance of the standard JPEG 2000 over all tested images.

References

[1] C. Perra, "Lossless plenoptic image compression using adaptive block differential prediction," in *ICASSP*, April 2015, pp. 1231–1234.

[2] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *CVPR*, June 2013, pp. 1027–1034.

[3] P. Helin, P. Astola, B. Rao, and I. Tabus, "Sparse modelling and predictive coding of subaperture images for lossless plenoptic image compression," in *3DTV-CON*, July 2016.

[4] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Transactions on Signal Processing*, vol. 43, no. 7, pp. 1713–1715, July 1995.

[2] Available at http://mmspg.epfl.ch/EPFL-light-field-image-dataset.

ISBN 978-0-9928626-7-1 © EURASIP 2017