Sparse modeling for efficient region and object based plenoptic image compression

Ioan Tabus[⋆]

Joint work with Petri Helin[⋆], Pekka Astola[⋆], Bhaskar Rao[†]

[⋆] Department of Signal Processing, Tampere University of Technology, Tampere, FINLAND
[†] Department of Electrical and Computer Engineering, University of California, San Diego, CA, USA

July 30, 2017

Recent work on sparse modelling for plenoptic camera image compression

1. 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.
2. I. Tabus and P. Helin, Microlens image sparse modelling for lossless compression of plenoptic camera sensor images, In Proceedings of EUSIPCO 2017, Kos, August 2017.
3. I. Tabus, P. Helin, P. Astola, "Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000", ICIP 2017, Beijing, September 2017.
4. P. Helin, P. Astola, B. Rao, and I. Tabus, "Minimum description length sparse modeling and region merging for lossless plenoptic image compression", Accepted with minor revisons, Journal Selected Topics on Signal Processing, 2017.

Plenoptic camera

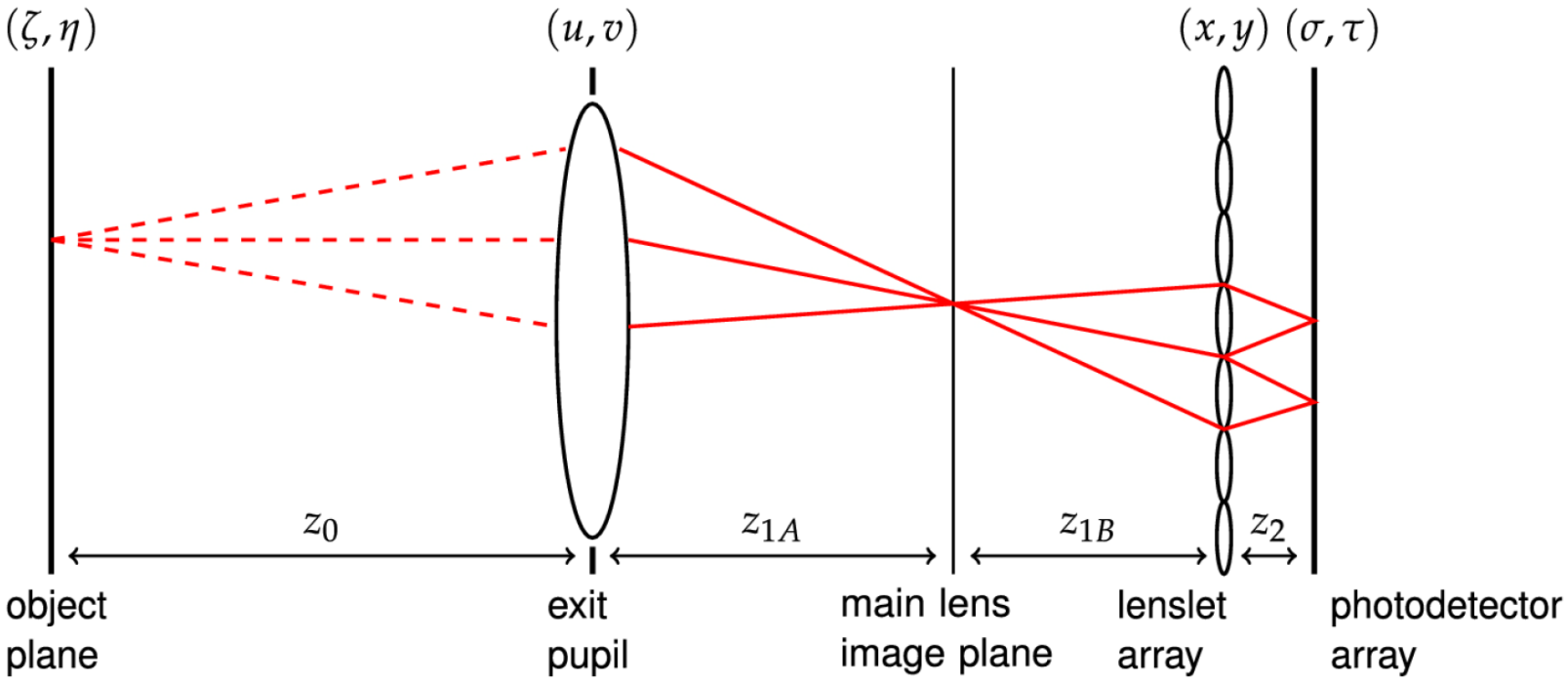


Figure 1: Focused plenoptic camera system. Object is focused at a plane in front of the lenslet array, and the microlenses then act as a relay system with the main camera lens.

Plenoptic camera (Lytro Illum)

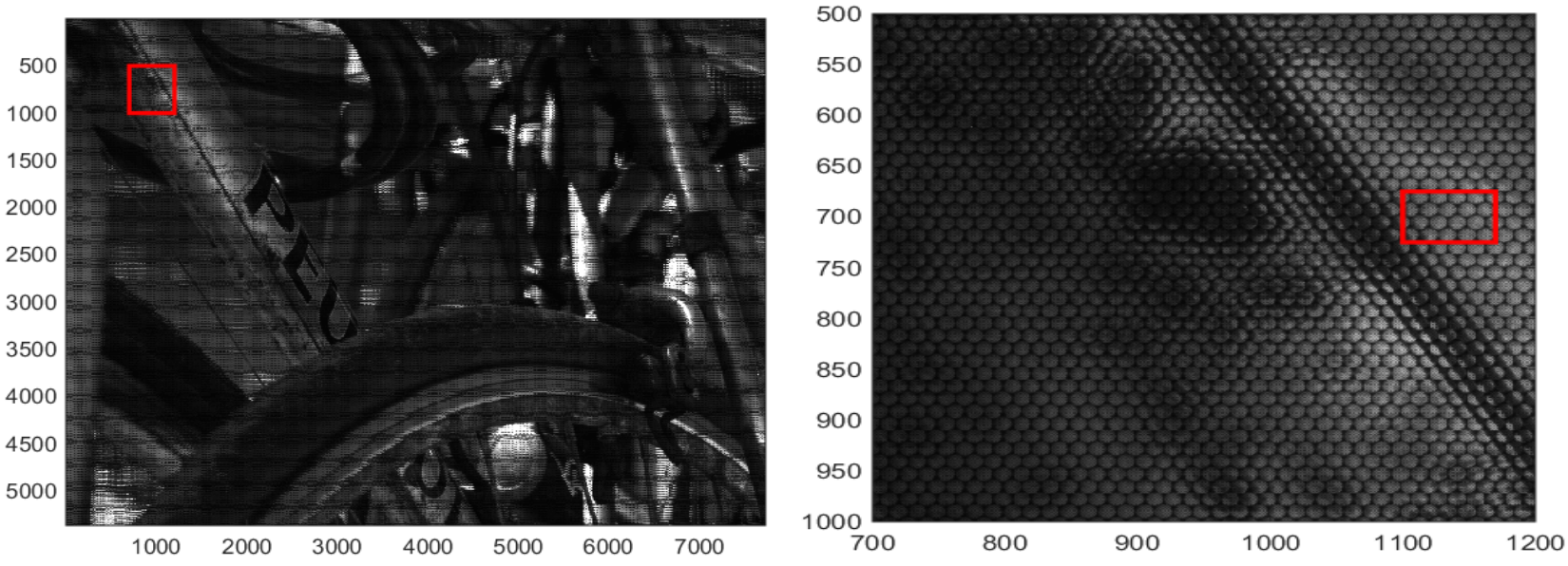
Focusing at different depths

(a) (b) (c)

Plenoptic camera (Lytro Illum) images after postprocessing:

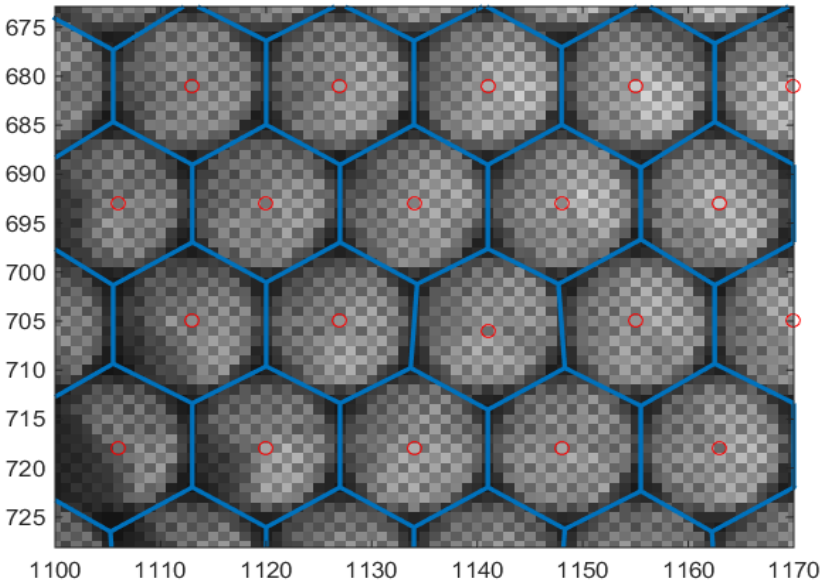
- Post-focusing on the foreground objects
- Post-focusing on the mid-depth objects
- Depth map obtained combining epipolar-plane line matching and depth-from-defocus

Image captured by the sensor

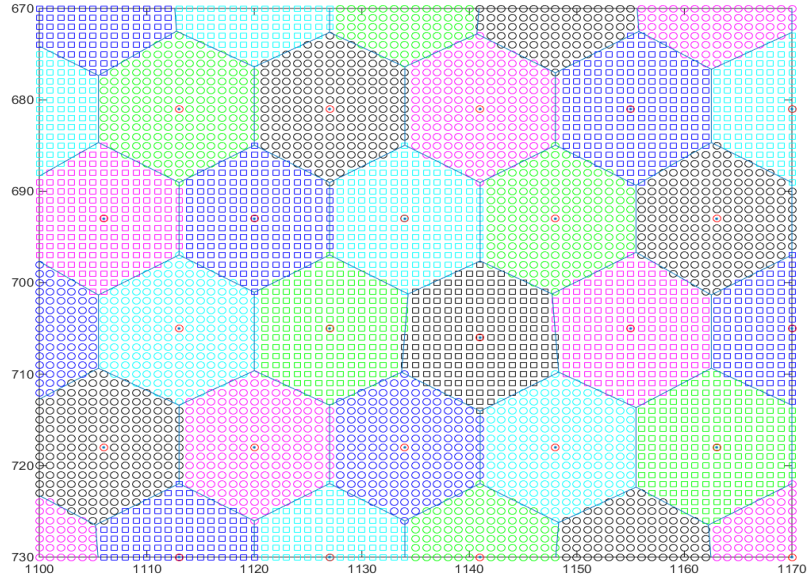


- Microlens array $(434 \times 541) = 234794$ microlenses
- Full sensor image: (5368×7728) pixels; (434×541) macropixels
- Zoom revealing lenslet structure

Image captured by the sensor



(a)



(b)

(a) the hexagonal grid of microlense centers is estimated with the camera meta-parameters Ψ

- the centers have integer coordinates marked by red circles;
- the Voronoi cells determined by the lattice of centers are drawn with blue lines;

(b) 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.

First problem: lossless compress the row sensor image (Eusipco 2017)

- First problem is the lossless compression 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.

Spiral scanning of a microlens block for marking the directions of views

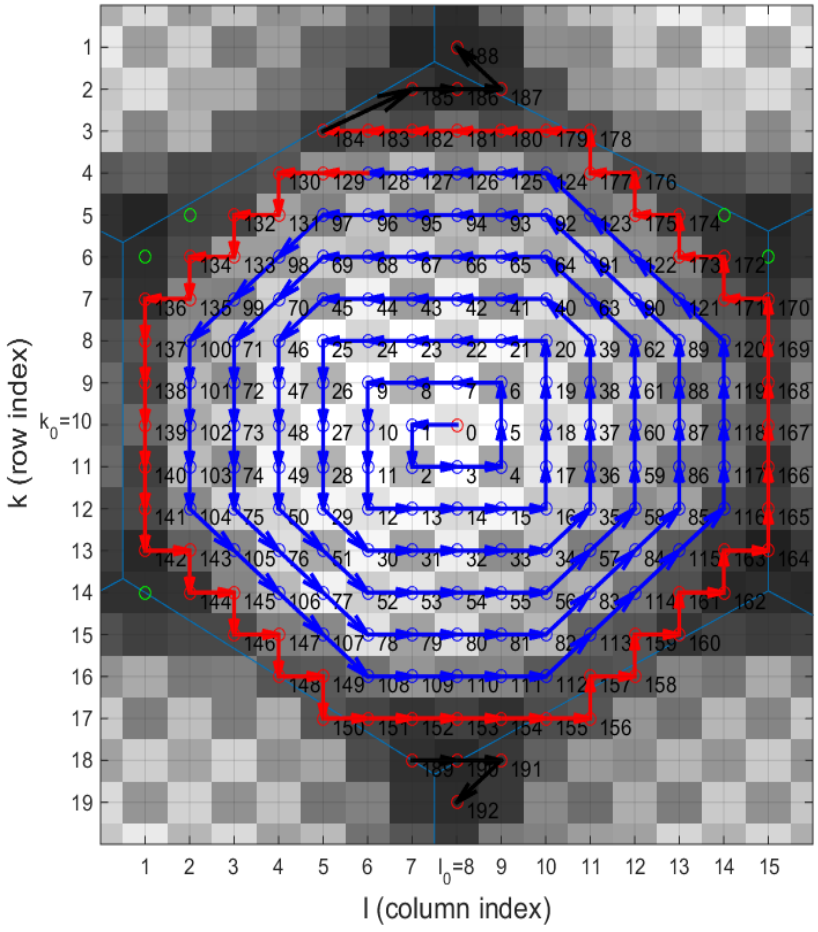


Figure 2: Spiral scanning of a microlens block. The template $\mathbf{T}$ alocates an index $T(k,l) \in \{0,\ldots 192\}$ to the pixel at (k,l), e.g. $T(10,8) = 0$ and $T(10,7) = 1$. The template selects 193 pixels. The elements of each microlens block (i,j) selected by the template are forming the 192-dimensional vector $\mathbf{v}^{(i,j)}$.

Lossless compression using frame prediction

Linear predictive coding

- The signal to be encoded is $\mathbf{x} = [x_0 \ \ldots \ x_{n-1}]^T$.
- For each sample x_i we also know a regressor vector $\boldsymbol{\psi_i}$ of size m (however some entries of $\boldsymbol{\psi_i}$ may be irrelevant).
- Linear prediction is obtained by

$$\hat{x}_i = \boldsymbol{\psi}_i^T \mathbf{w},$$

  where the data matrix is $\Psi = [\boldsymbol{\psi}_0 \ \ldots \ \boldsymbol{\psi}_{n-1}]^T$.
- The vector of predictions $\hat{\mathbf{x}} = [\hat{x}_0 \ \ldots \ \hat{x}_{n-1}]^T$ is given by $\hat{\mathbf{x}} = \Psi \mathbf{w}$.
- The prediction residuals are forced to be integer valued by rounding the linear predictions $\hat{x}_i$ to their nearest integer, $\lfloor \hat{x}_i \rceil$, and hence the definition of the prediction residual vector $\boldsymbol{\epsilon} = [\epsilon_0 \ \ldots \ \epsilon_{n-1}]^T$ is

$$\boldsymbol{\epsilon} = \mathbf{x} - \lfloor \hat{\mathbf{x}} \rceil = \mathbf{x} - \lfloor \Psi \mathbf{w} \rceil \, ;$$

- The encoder will transmit the parameter vector $\mathbf{w}$ and the prediction residuals $\boldsymbol{\epsilon}$, for lossless reconstruction.

Lossless compression using sparse prediction

Linear predictive coding

- The residual encoding algorithm $\mathcal{A}_\epsilon$ uses $\mathcal{L}(\boldsymbol{\epsilon}|\mathcal{A}_\epsilon)$ bits
- Each predictor coefficient w_j is truncated to b bits; hence it belongs to the set $\mathcal{Q}_b = \{i \cdot 2^{-b} : i \in \mathbb{Z}\}$ for a fixed $b \in \{0, \ldots, 16\}$.
- The encoding algorithm $\mathcal{A}_w$ encodes $\mathbf{w} \in \mathcal{Q}_b^m$ using $\mathcal{L}(\mathbf{w}|\mathcal{A}_w)$ bits. The decoder can reconstruct the original samples by

$$\mathbf{x} = \boldsymbol{\epsilon} + \lfloor \Psi \mathbf{w} \rceil ,$$

  where the needed samples in the ith row of the matrix Ψ are already decoded and are available at the time of computing the entry x_i.
- The optimization problem P_F to be solved at the encoder for each frame is

$$P_F : \min_{\mathbf{w}} \mathcal{L}(\boldsymbol{\epsilon}|\mathcal{A}_\epsilon) + \mathcal{L}(\mathbf{w}|\mathcal{A}_w) \text{ s.t. } \boldsymbol{\epsilon} = \mathbf{x} - \lfloor \Psi \mathbf{w} \rceil ; \ \mathbf{w} \in \mathcal{Q}_b^m,$$

  encompassing finding the optimal predictor parameters and the optimal structure of the predictor.
- Most of existing methods use this scheme. There is a high variety of encoding algorithms $\mathcal{A}_\epsilon$, $\mathcal{A}_w$, and of optimization method for solving P_F.

Lossless compression using sparse prediction

Approximating the expressions of the codelengths for residuals and prediction coefficients

- We can remove rounding when solving the optimal problem (but the rounding needs to be effectively applied at both encoder and decoder). The new form of the problem becomes:

$$P_{F_1}: \min_{\mathbf{w}} \mathcal{L}(\boldsymbol{\epsilon}|\mathcal{A}_\epsilon) + \mathcal{L}(\mathbf{w}|\mathcal{A}_w) \text{ s.t. } \mathbf{x} = \Psi\mathbf{w} + \boldsymbol{\epsilon};\ \mathbf{w} \in \mathcal{Q}_b^m.$$

- We assume that the residuals have a Gaussian distribution, resulting in $\mathcal{L}(\boldsymbol{\epsilon}|\mathcal{A}_\epsilon) = \frac{n}{2}\log_2\sum_{i=0}^{n-1}\epsilon_i^2 + \frac{n}{2}\log_2\frac{2\pi e}{n}$, leading to the following particular case of the optimization problem P_{F_1}:

$$P_{F_2}: \min_{\mathbf{w}} \frac{n}{2}\log_2(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}) + \mathcal{L}(\mathbf{w}|\mathcal{A}_w) \text{ s.t. } \mathbf{x} = \Psi\mathbf{w} + \boldsymbol{\epsilon};\ \mathbf{w} \in \mathcal{Q}_b^m.$$

- The sparsity pattern $\boldsymbol{\gamma}$ is explicitly coded as part of the intermediate variables used in the algorithm $\mathcal{A}_w$ and hence $\mathcal{L}(\mathbf{w}|\mathcal{A}_w)$ will be mainly a function $f(\boldsymbol{\gamma})$ leading to a more particular form for the problem P_{F_2}:

$$P_{F_3}: \min_{\mathbf{w}} \frac{n}{2}\log_2(\boldsymbol{\epsilon}^T\boldsymbol{\epsilon}) + f(\boldsymbol{\gamma}) \text{ s.t. } \mathbf{x} = \Psi\mathbf{w} + \boldsymbol{\epsilon};\ \mathbf{w} \in \mathcal{Q}_b^m.$$

Greedy optimization algorithm for sparse prediction

Algorithm A *Optimization algorithm for the encoding step E1.*

A0. Given m, k^{max}, Ψ, and $\mathbf{x}$. Set $\gamma_i^* = 0$ for all $i = 0, \ldots, m-1$ and define $\mathcal{C}_{[0]} = \{0, \ldots, m-1\}$ the set of zero positions in $\boldsymbol{w}$.

A1. For $k = 1, \ldots, k^{max}$

A1.1 For $i \in \mathcal{C}_{[k-1]}$

A1.1.1 Set $\boldsymbol{\gamma} = \boldsymbol{\gamma}^*$ and $\gamma_i = 1$.

A1.1.2 Compute *efficiently* $\|\mathbf{x} - \Psi_\gamma \hat{\boldsymbol{\theta}}\|_2^2 = \min_{\boldsymbol{\theta}} \|\mathbf{x} - \Psi_\gamma \boldsymbol{\theta}\|_2^2$ and store the current criterion $J_i = \frac{n}{2} \log_2 \|\mathbf{x} - \Psi_\gamma \hat{\boldsymbol{\theta}}\|_2^2 + f(\boldsymbol{\gamma})$. EFFICIENT: OMP, or OOMP, or LARS sparse algorithms

A1.2 Take $i^* = \arg\min_i J_i$, set $\gamma_{i^*}^* = 1$, $\mathcal{C}_{[k]} = \mathcal{C}_{[k-1]} \setminus \{i^*\}$, and store the solution at step k: $\boldsymbol{\gamma}_{[k]} = \boldsymbol{\gamma}^*$.

A1.3 For the retained $\boldsymbol{\gamma}^*$ find $\hat{\boldsymbol{\theta}}$ as solution of

$$P_\gamma : \min_{\boldsymbol{\theta}} \boldsymbol{\epsilon}^T \boldsymbol{\epsilon} \text{ s.t. } \mathbf{x} = \Psi_{\gamma^*} \boldsymbol{\theta} + \boldsymbol{\epsilon};\ \boldsymbol{\theta} \in \mathcal{Q}_b^{\|\gamma^*\|_0},$$

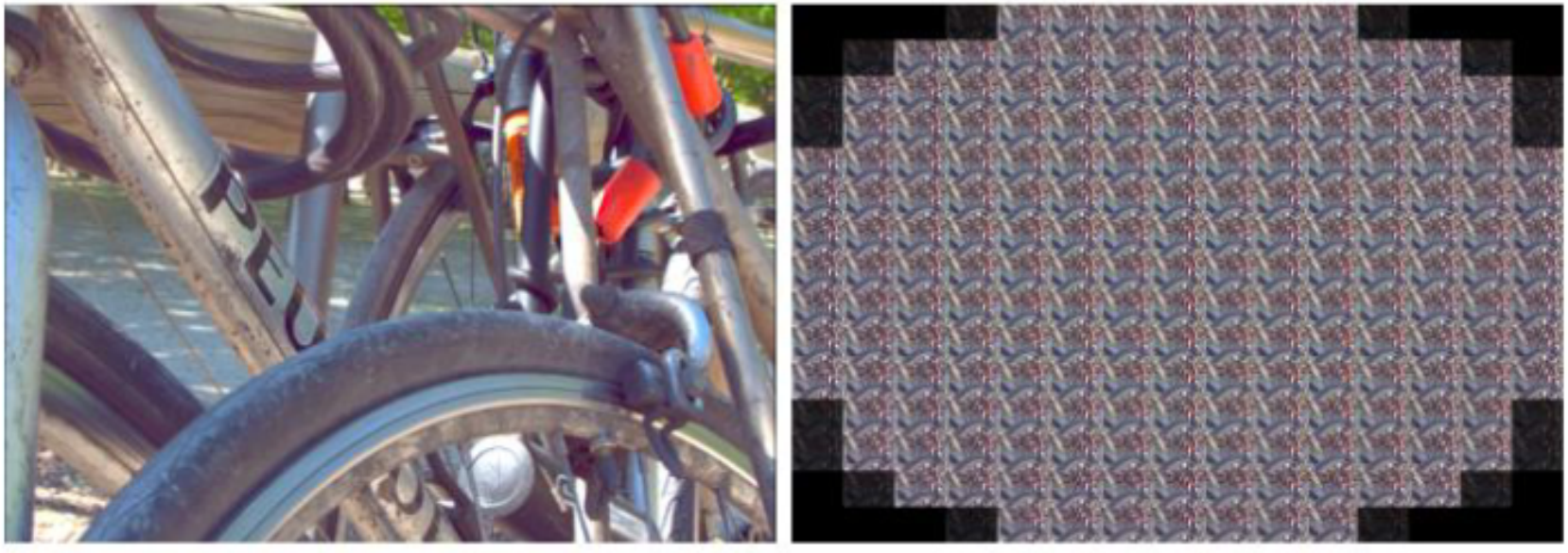
denote $\mathbf{w}_{[k]}$ the vector with structure $\boldsymbol{\gamma}^*$ and nonzero entries $\hat{\boldsymbol{\theta}}$, and store the criterion $C_{[k]} = \frac{n}{2} \log_2 \|\mathbf{x} - \Psi_{\gamma^*} \hat{\boldsymbol{\theta}}\|_2^2 + \mathcal{L}(\mathbf{w}_{[k]} | \mathcal{A}_w)$.

A2. Take the winning stage as $k^* = \arg\min_k C_{[k]}$ and output $\mathbf{w}_{[k^*]}$.

Detailed descriptions

- Audio signals
  - F. Ghido I. Tabus. "Sparse modeling for lossless audio compression". *IEEE Transactions on Audio, Speech, and Language Processing*, 21:1, pp. 14-28, 2013.
  - B. Dumitrescu, A. Onose, P. Helin, I. Tabus. "Greedy Sparse RLS". *IEEE Transactions on Signal Processing*, 60:5, pp. 2194–2207, 2012.
- Plenoptic images
  - P. Helin, P. Astola, B. Rao, and I. Tabus, "Sparse modelling and predictive coding of subaperture images for lossless plenoptic image compression", in 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), pp.1-4, Hamburg, Germany, 4-6 July 2016.

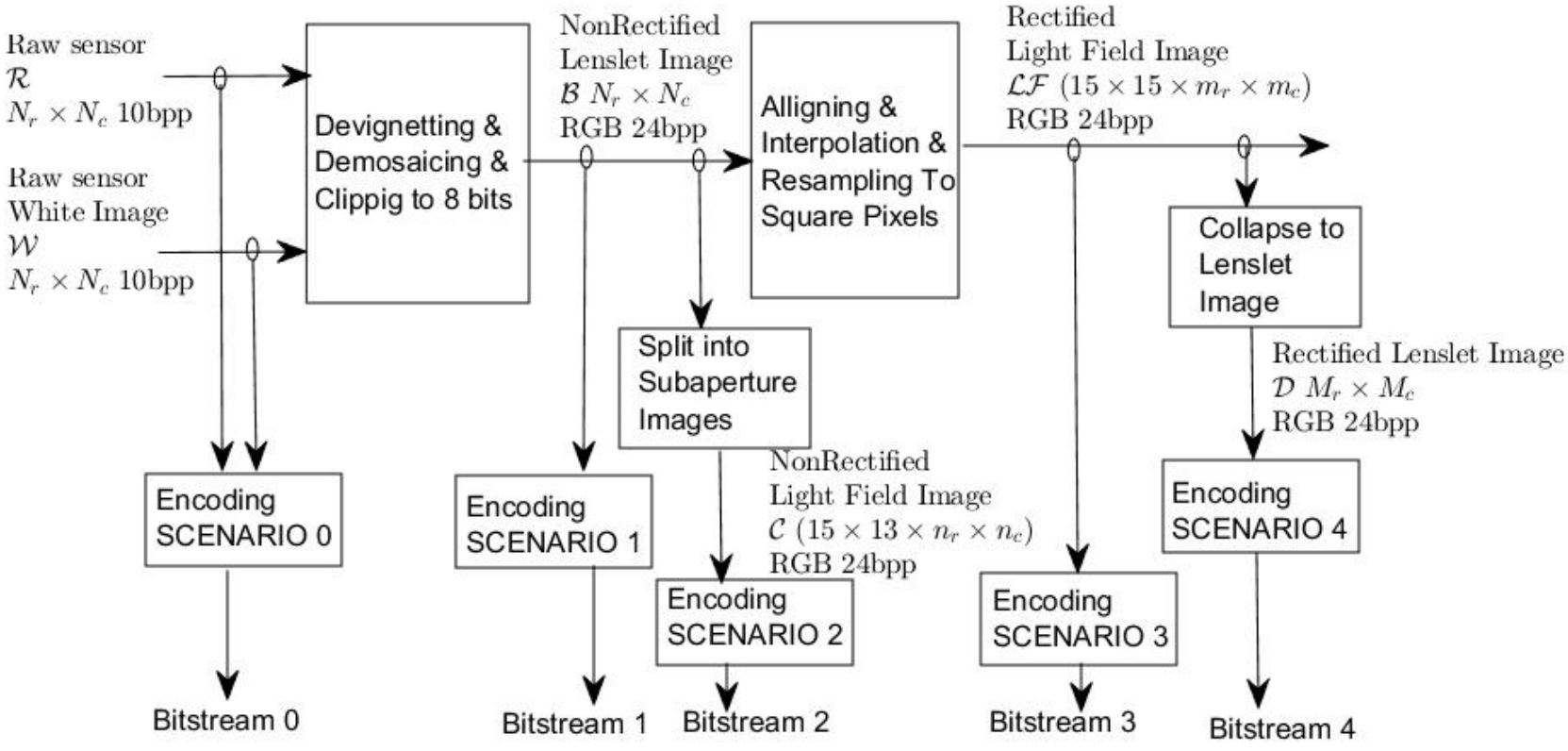
Plenoptic camera (Lytro Illum) images after postprocessing



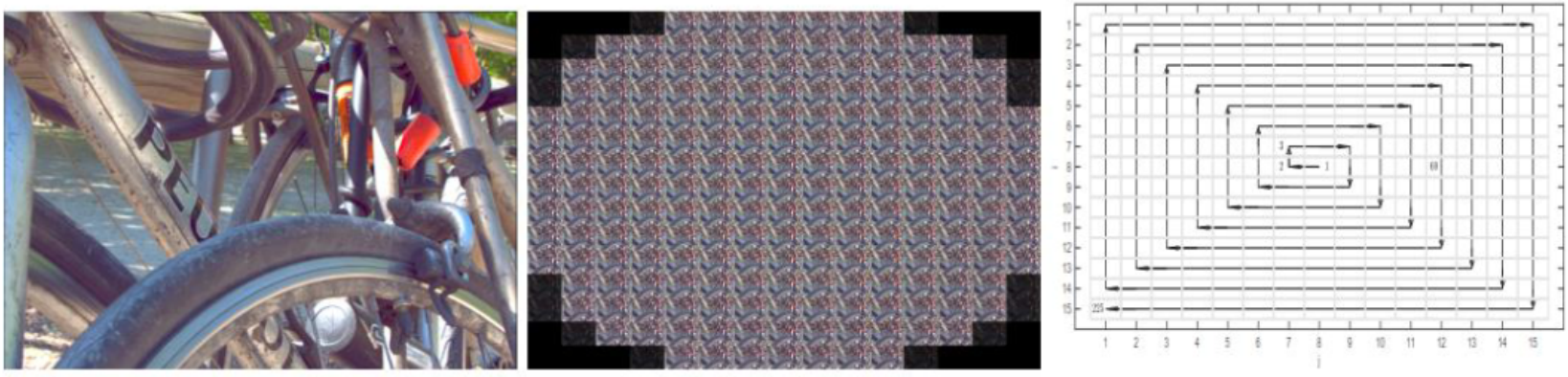
(a) (b)

(a) One view consisting of a 434×625 color image
(b) 15×15 array of views, each view being a 434×625 color image

Processing chain of a plenoptic image



Plenoptic camera (Lytro Illum) images after postprocessing



(a) (b) (c)

Compressing the LF structure

Problem formulation

- An image captured with a plenoptic camera is processed in a chain consisting of demosaicing, devignetting, alignment and rectification.
- The result is a 5D light-field structure $L_F(i, j, x, y, c)$
  - $i, j \in \{1, \ldots, N\}$ are indices for a certain pixel under a microlens (hence addressing a particular subaperture image)
  - $x \in \{1, \ldots, \xi\}, y \in \{1, \ldots, \eta\}$ correspond to the spatial pixel location in the subaperture image
  - there are NxN subaperture images (views) A_{ij}
- Experiments with plenoptic images from Lytro Illum, where $N = 15, \xi = 434$ and $\eta = 625$.
- The view with index $(i_c, j_c) = (8, 8)$ is called center view and the other views side views.
- The task is to compress simultaneously Nx$N = 225$ views losslessly, where the close-by views are extremely similar.

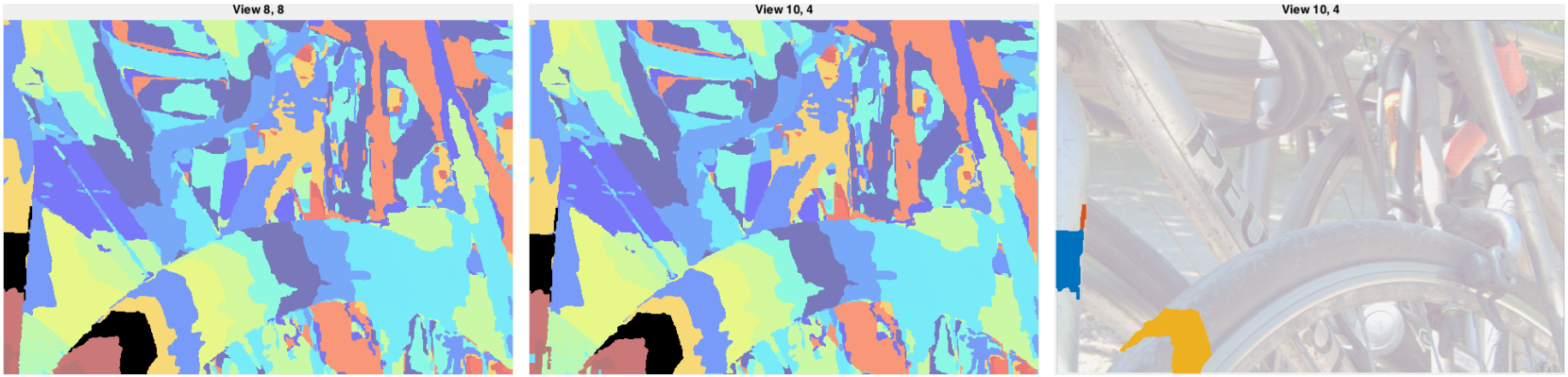
Proposed scheme

Method overview

- A predictive scheme where one optimal sparse predictor is designed for each region from a segmentation of the side view.
- The segmentations of all side views are based on a quantized depth map of the center view.
  - Obtained in the first stage, by propagating the center view regions using disparities along x and y coordinates and suitably filling the missing regions that may result after warping.
- One region of a side view is predicted using an optimal sparse predictor having as regressors pixels from several side views, accounting for small integer or fractional disparities.
  - Done sequentially over all side views.
  - The center RGB view (compressed with JPEG 2000) and the quantized depth map are sent to the decoder.

Method

Partition propagation



(a) Regions in center view. (b) Propagated regions in the side view. (c) Regions at depth $Z = 3$.

Figure 3: a) Segmentation of the depth map of the center view in $K = 32$ quantization levels. The three connected components (regions) forming the region $\Omega^3_{8,8}$ at depth $Z = 3$ are shown in black. b) Propagation of the depth segments to the side view $(10, 4)$. Region $\Omega^3_{10,4}$ is shown in black. c) Location of the three connected component regions: $R^{3,0}_{10,4}$(orange), $R^{3,1}_{10,4}$ (blue), and $R^{3,2}_{10,4}$ (red) overlaid over the color side view image.

Method overview

Partition propagation

- First, the partition of a side view $\{\Omega_{ij}^0, \ldots, \Omega_{ij}^{K-1}\}$ is obtained using the partition of the central view $\{\Omega_{i_c j_c}^0, \ldots, \Omega_{i_c j_c}^{K-1}\}$ (having the label image $Z_{i_c j_c}$).
  - For each region the disparities are sought by minimizing the Mean Square Error (MSE) in the color views:

$$(\delta_{ij}^k, \mu_{ij}^k) = \arg\min_{\delta,\mu} \sum_{x,y \in \Omega_{i_c j_c}^k} \sum_{c=0}^{2} \left(A_{i_c j_c}(x, y, c) - A_{ij}(x + \delta, y + \mu, c)\right)^2. \quad (1)$$

  - Using these disparities, the constant depth regions of the center view are warped to the side view.
  - The warping operation may create a set of missing pixels due to dis-occlusions, denoted by Φ.
    - The missing pixels are treated by allocating them to the nearby region (with respect to the coordinates x and y) located at the farthest depth, as described in steps 2.2–2.4 in Alg. 1.

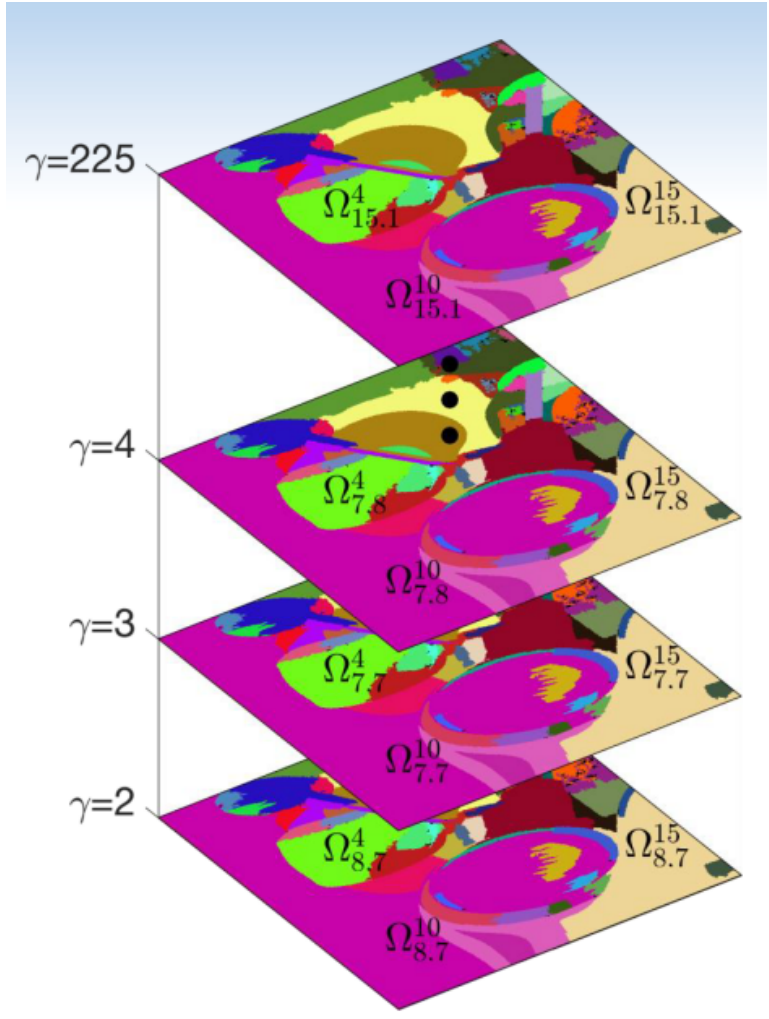
Algorithm 1 Partition propagation from central view (i_c, j_c) to side view (i, j).

Input: Depth map for the center view $Z_{i_c j_c}$ which labels the partition $\{\Omega^0_{i_c j_c}, \ldots, \Omega^{K-1}_{i_c j_c}\}$; disparities δ^k_{ij} and μ^k_{ij}, for $k = 0, \ldots, K-1$.

1. For $k = K-1, \ldots, 0$
   For each $(x, y) \in \Omega^k_{i_c, j_c}$
   1.1 Set $x' = x + \delta^k_{ij}$ and $y' = y + \mu^k_{ij}$.
   1.2 Insert (x', y') into Ω^k_{ij}.
2. Fill missing pixels $\Phi = \Omega \setminus \{\Omega^0_{ij}, \ldots, \Omega^{K-1}_{ij}\}$ to form partition:
   For $k = K-1, \ldots, 0$:
   2.1 Dilate the set Ω^k_{ij} with a disk of radius r to obtain region Φ_1.
   2.2 $\Phi_2 = \Phi \cap \Phi_1$ (pick from dilated region only missing pixels).
   2.3 $\Omega^k_{ij} = \Omega^k_{ij} \cup \Phi_2$ (allocate to set Ω^k_{ij}).
   2.4 $\Phi = \Phi \setminus \Phi_2$ (discard Φ_2 from missing pixels).

Output: $\{\Omega^0_{ij}, \ldots, \Omega^{K-1}_{ij}\}$.

Segmentation propagation



Utilizing view cross-correlation

- Denote Γ_{ij} the set of pairs (i', j') indexing the views that are available when designing a predictor for the view A_{ij}
- Limit the cardinality Γ so that $|\Gamma| \leq 5$ by choosing the nearest views i.e. the views (i', j') for which $|i - i'| + |j - j'|$ is smallest.

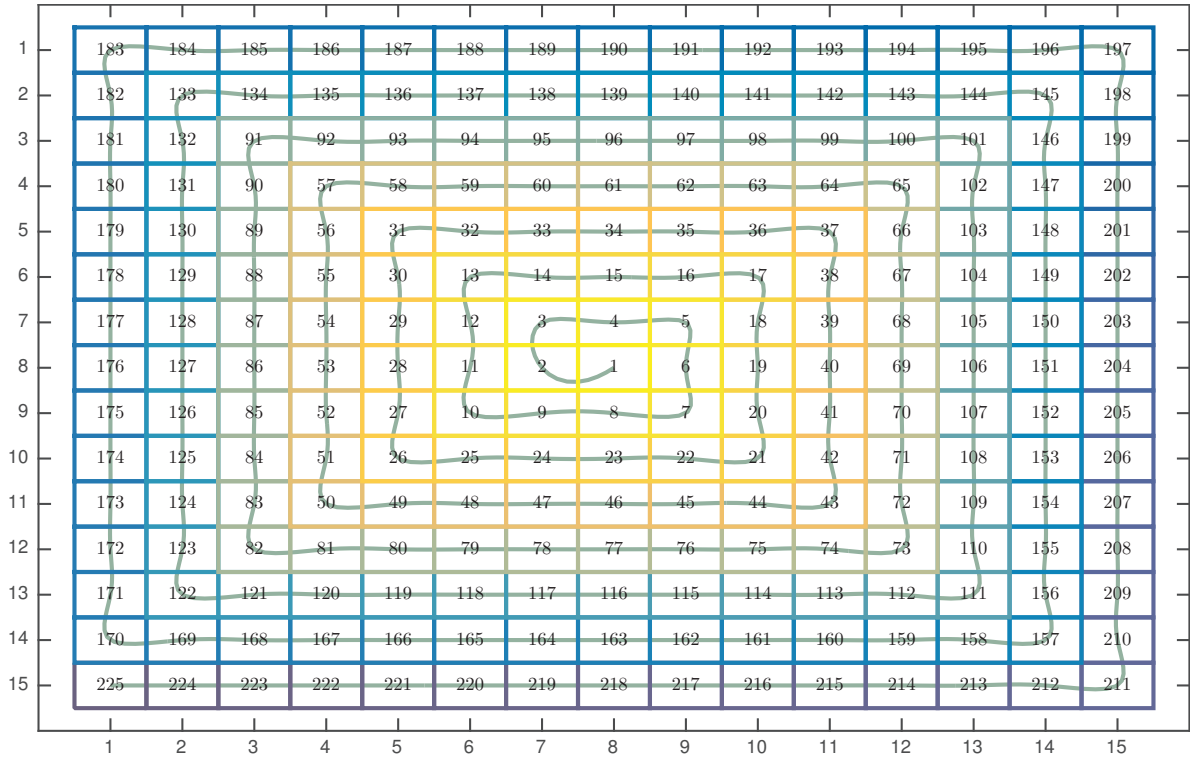
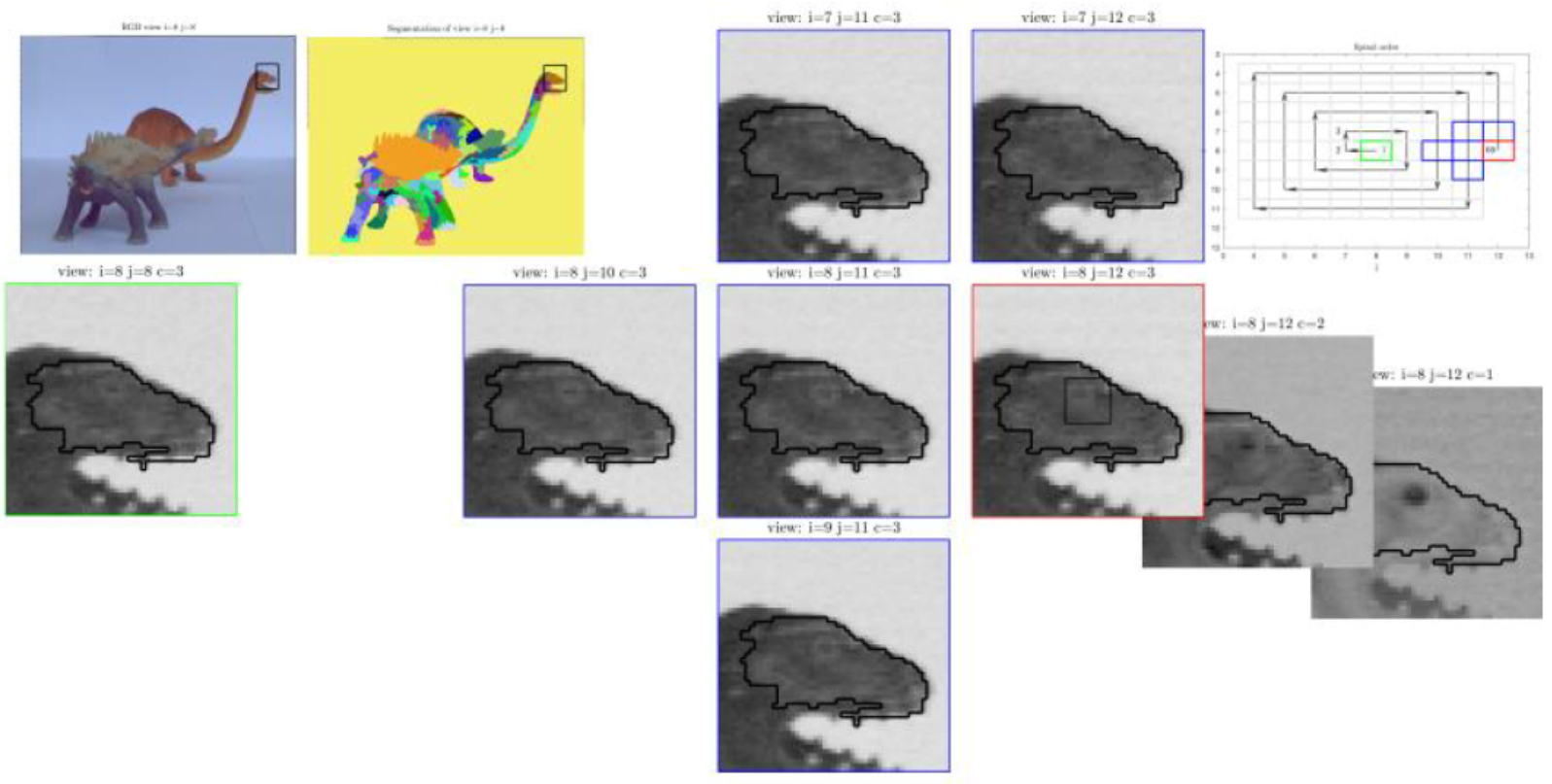


Figure 4: The processing order of the views. The index inside the rectangles gives the processing order number while the vertical axis represents i and horizontal axis represents j.

Utilizing view cross-correlation



Method

Encoding scheme

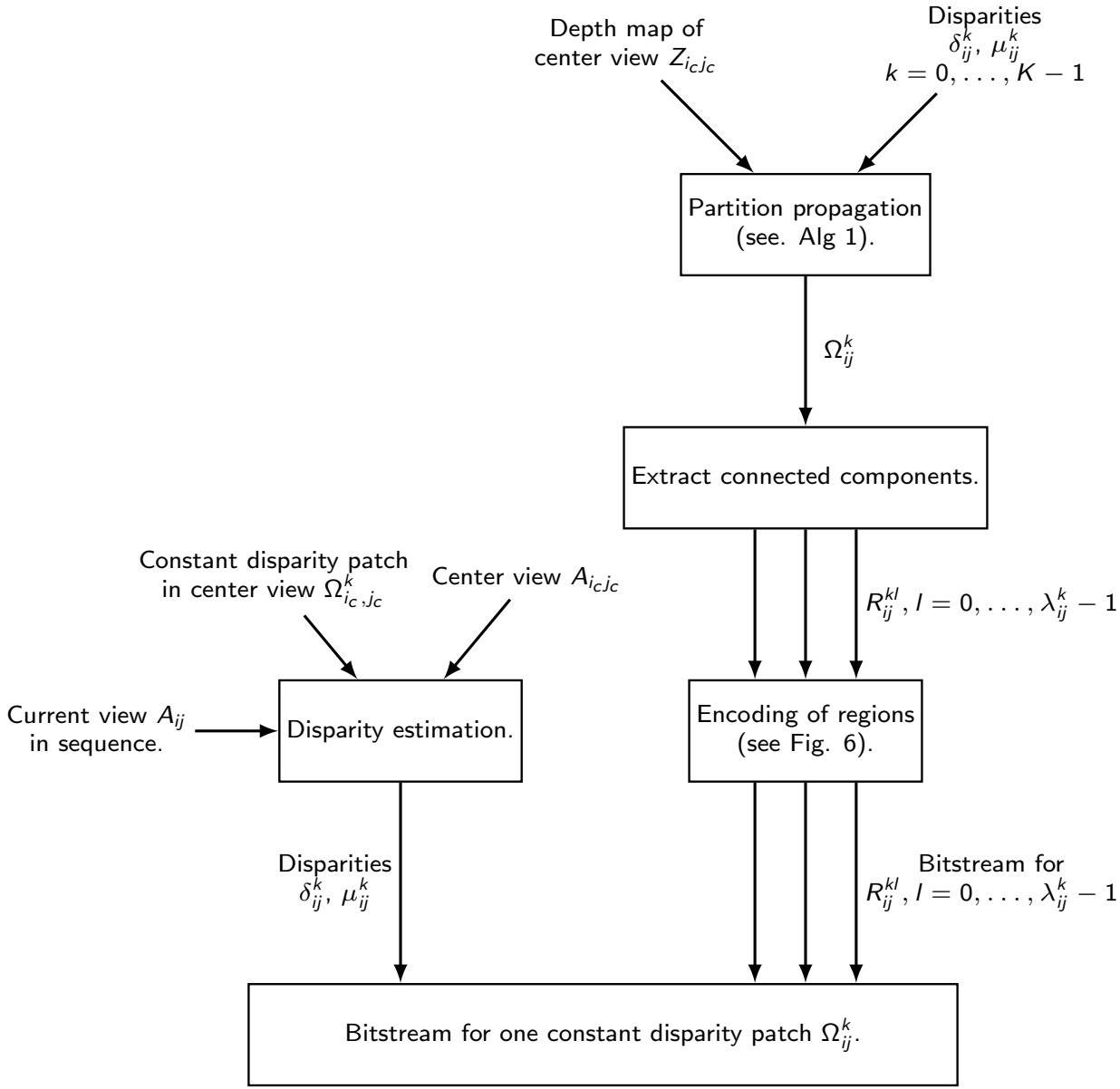


Figure 5: Encoding a constant disparity patch Ω_{ij}^k in sideview (i, j).

Method

Encoding scheme

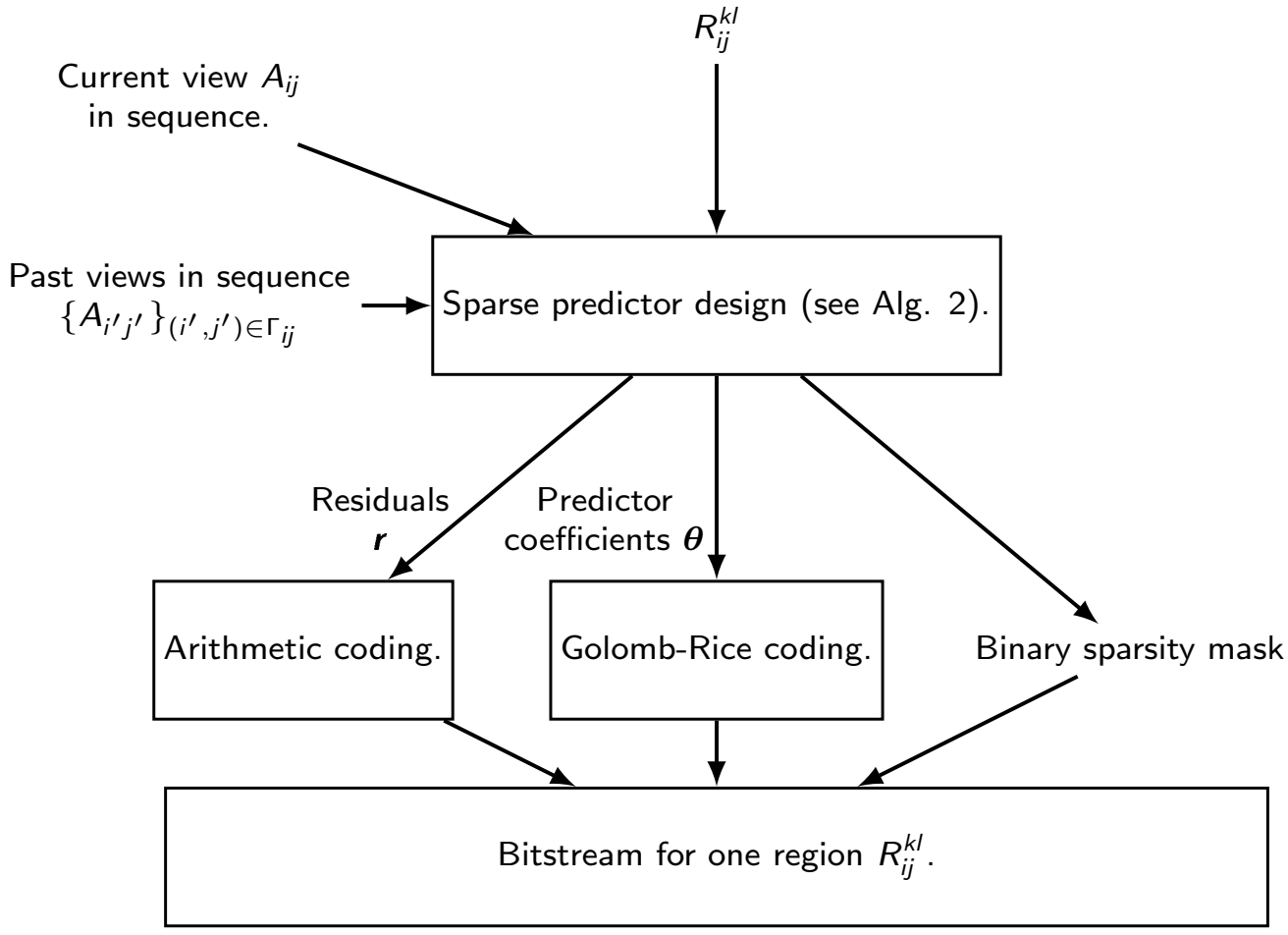


Figure 6: Encoding the region R_{ij}^{kl} in side view (i, j).

Sparse modelling

Sparse prediction model

- For the region R_{ij}^{kl} we introduce the sparse modeling problem as

$$\text{minimize}_{\boldsymbol{\theta}} \quad \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{\theta}\|_2 \quad \text{s.t.} \quad \|\boldsymbol{\theta}\|_0 \leq \kappa \tag{2}$$

- A row of the desired vector $\boldsymbol{y}$:

$$y_r = A_{ij}(x, y, c). \tag{3}$$

  where $(x, y) \in R_{ij}^{kl}$.

- The elements of the r-th row of the matrix $\boldsymbol{D}$:
  1. The constant element 1 compensating for the bias.
  2. The N,W,NW and NE causal neighbors of (x, y) from the current view (e.g. the north element is $A_{ij}(x, y-1, c)$).
  3. For all views $(i', j') \in \Gamma_{ij}$:

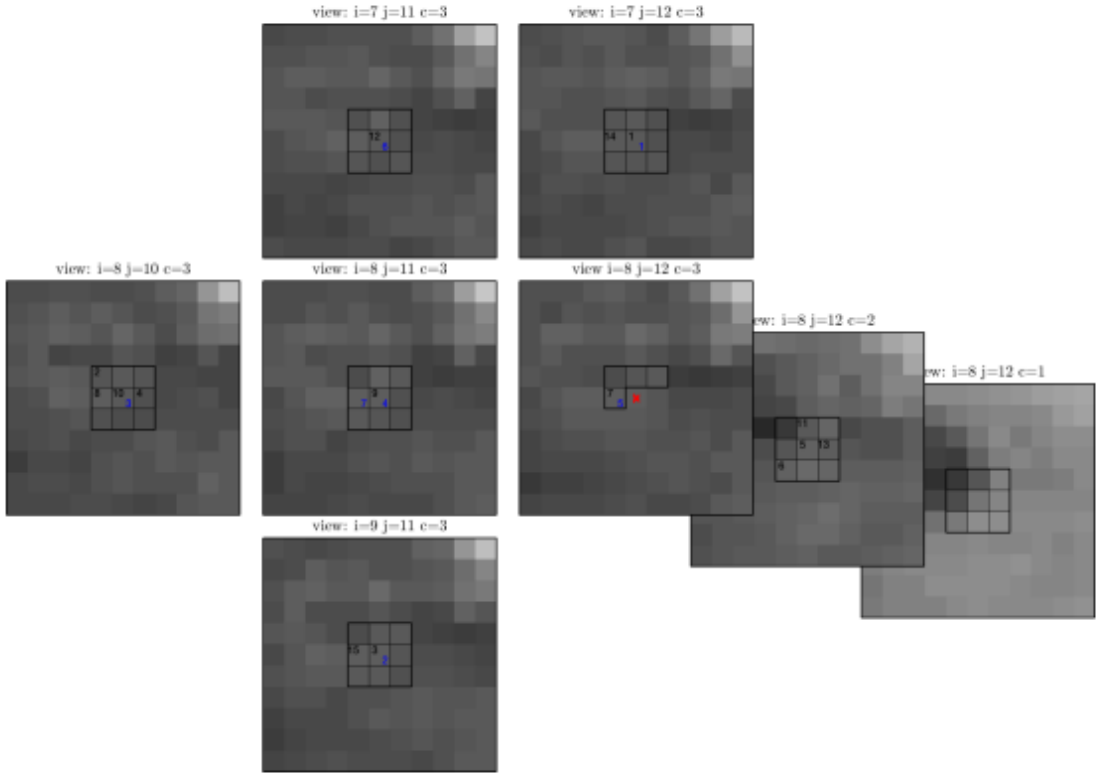
     A row vector consisting of the elements of $A_{i'j'}$ from the full 3x3 neighborhood of (x, y) (e.g. if we denote $C = A_{i'j'}$, the row vector is

$$[\, C_{x,y} \;\; C_{x-1,y} \;\; C_{x-1,y-1} \;\; C_{x-1,y+1} \;\; C_{x,y-1}$$
$$C_{x,y+1} \;\; C_{x+1,y-1} \;\; C_{x+1,y} \;\; C_{x+1,y+1} \,]$$

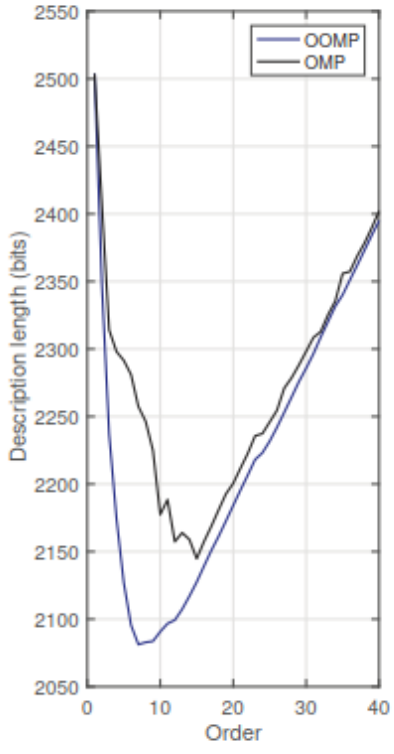
     where we omitted the index for color and denote as subscripts the pixel indices).

Sparse modelling

Regression matrix



(a) The available regressors and selected ones by OMP and OOMP.



(b) Description length for the region.

Figure 7: The regressors. An (11 11)-square where all pixels belong to the same region is shown in subgures (a)-(b). (a) Illustration of OOMP template chosen by MDL. The pixel to be predicted, marked as a red in view i = 8 j = 12 c = 3, is set as the t'th element of the vector y. Its causal neighbors inside the black border regions are elements of the t'th row of candidate regressors matrix D, together with all pixels inside the black contours.

Algorithm 2 Sparse prediction.

Input: desired vector $\boldsymbol{y}$, regressors $\boldsymbol{D}$, maximum order of prediction $\kappa_{\max}$.
Initialize $\kappa = 1$, the set of selected indices $\mathcal{I}_0 = \emptyset$ and the residuals $\boldsymbol{r} = \boldsymbol{y}$.
1. For each column of $\boldsymbol{D}$, denoted $\boldsymbol{d}_i$, compute the criterion

$$c_i = \frac{|\boldsymbol{r}^T \boldsymbol{d}_i|}{\|\boldsymbol{d}_i\|}$$

2. Append the index i that maximizes the criterion to the set of selected indices $\mathcal{I}_\kappa = \mathcal{I}_{\kappa-1} \cup \arg\max_i c_i$.
3. Compute the nonzero values of the predictor $\boldsymbol{\theta}_\kappa = \boldsymbol{D}^\dagger_{\mathcal{I}_\kappa} \boldsymbol{y}$ where $\boldsymbol{D}^\dagger$ is the pseudoinverse of $\boldsymbol{D}$.
4. Quantize the coefficients to 10 bits.
5. Compute the code length $\mathrm{CL}_{\boldsymbol{r}} + \mathrm{CL}_{\mathrm{mask}} + \mathrm{CL}_{\boldsymbol{\theta}_\kappa}$.
6. If $\kappa < \kappa_{\max}$, set $\kappa = \kappa + 1$ and go to step 1.
7. Compute $\hat{\kappa}$ as in (**??**).

Output: The nonzero coefficients $\boldsymbol{\theta}_{\hat{\kappa}}$, and the sparsity mask corresponding to $\mathcal{I}_{\hat{\kappa}}$.

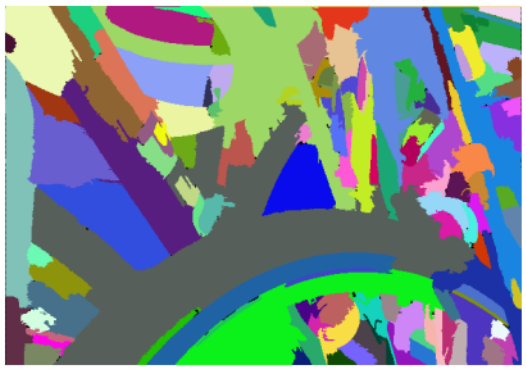
Results

- Implementation with a combination of Matlab and C. Also the decoder is implemented.
- Subset of images from EPFL Light-Field Image Dataset.
- The size of the raw 24 bit data is 183 MB.

Table 1: Compressed file sizes in mega bytes (MB).

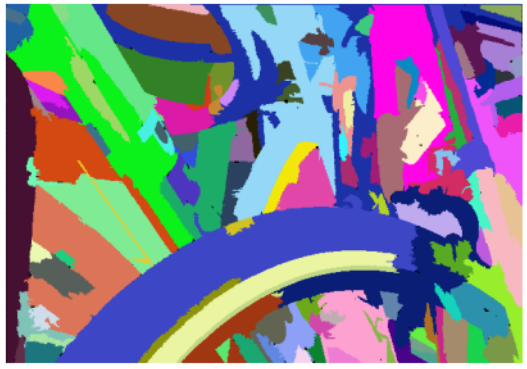
i_l	HEVC	HM	AVC	CALIC	FP8	[3DTV16]	OOMP-CL-RM-1
1	81.82	88.32	80.61	82.47	82.99	69.66	**62.85**
2	87.04	95.46	86.36	88.63	86.75	77.06	**67.06**
3	86.65	96.25	86.19	88.65	86.06	78.14	**68.18**
4	81.50	87.75	80.18	81.30	78.76	67.28	**61.82**
5	84.33	92.38	82.71	87.94	83.23	71.90	**66.20**
6	70.86	77.79	69.74	75.82	68.40	58.99	**53.63**
7	56.47	72.77	55.87	63.45	52.86	50.79	**47.29**
8	70.74	77.21	69.07	75.95	67.17	59.56	**54.54**
9	85.51	92.67	84.15	85.75	85.67	70.20	**64.26**
10	69.46	77.76	67.58	75.20	66.92	60.19	**53.51**
11	82.83	96.40	81.95	93.36	78.92	71.29	**65.53**
12	78.43	84.96	78.35	76.85	75.04	71.40	**61.57**

Segmentations using MDL based region merging



(a) Color segmentation, -C.

(b) Constant disparity segmentation, -D.



(c) Constant disparity with region merging, -D-RM.

P. Helin, P. Astola, B. Rao, and I. Tabus, "Minimum description length sparse modeling and region merging for lossless plenoptic image compression", To be published in Journal of Selected Topics on Signal Processing, 2017.

Thank you!


Xiao Xiao, Bahram Javidi, Manuel Martinez-Corral, and Adrian Stern,
"Advances in three-dimensional integral imaging: sensing, display, and applications [invited],"
Appl. Opt., vol. 52, no. 4, pp. 546–560, Feb 2013.


T. Ebrahimi,
"Jpeg pleno abstract and executive summary,"
in *ISO/IEC JTC 1/SC 29/WG1 N6922*, Sydney, Australia, 2015.


Caroline Conti, Lus Ducla Soares, and Paulo Nunes,
"Hevc-based 3d holoscopic video coding using self-similarity compensated prediction,"
Signal Processing: Image Communication, vol. 42, pp. 59 – 78, 2016.


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.


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.

I. Tabus, I. Schiopu, and J. Astola,
"Context coding of depth map images under the piecewise-constant image model representation,"
IEEE Trans. Image Process., vol. 22, no. 11, pp. 4195–4210, Nov 2013.

T. C. Wang, A. A. Efros, and R. Ramamoorthi,
"Occlusion-aware depth estimation using light-field cameras,"
in *ICCV*, Dec 2015, pp. 3487–3495.

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

A. Vieira, H. Duarte, C. Perra, L. Tavora, and P. Assuncao,
"Data formats for high efficiency coding of lytro-illum light fields,"
in *IPTA*, Nov 2015, pp. 494–497.

I. Tabus and P. Astola,
"Sparse prediction for compression of stereo color images conditional on constant disparity patches,"
in *3DTV-CON*, July 2014, pp. 1–4.