

 Open access • Proceedings Article • DOI:10.1109/ICME.2017.8019442

Steered mixture-of-experts for light field coding, depth estimation, and processing

— [Source link](#) 

Ruben Verhack, Thomas Sikora, Lieven Lange, Rolf Jongebroed ...+2 more authors

Institutions: Ghent University, Technical University of Berlin

Published on: 01 Jul 2017 - International Conference on Multimedia and Expo

Topics: Light field, Mixture model, Edge detection, Interpolation and Piecewise

Related papers:

- [Interpreting plenoptic images as multi-view sequences for improved compression](#)
- [Optimized inter-view prediction based light field image compression with adaptive reconstruction](#)
- [HEVC-based light field image coding with bi-predicted self-similarity compensation](#)
- [Light field image coding via linear approximation prior](#)
- [Lossy compression of lenslet images from plenoptic cameras combining sparse predictive coding and JPEG 2000](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/steered-mixture-of-experts-for-light-field-coding-depth-2g9709qznn>

STEERED MIXTURE-OF-EXPERTS FOR LIGHT FIELD CODING, DEPTH ESTIMATION, AND PROCESSING

Ruben Verhack^{*†}, Thomas Sikora[†], Lieven Lange[†], Rolf Jongebroed[†],
Glenn Van Wallendael^{*}, and Peter Lambert^{*}

^{*}Ghent University - imec - IDLab, Ghent, Belgium

[†]Technische Universität Berlin - Communication Systems Lab, Berlin, Germany

ABSTRACT

The proposed framework, called Steered Mixture-of-Experts (SMoE), enables a multitude of processing tasks on light fields using a single unified Bayesian model. The underlying assumption is that light field rays are instantiations of a non-linear or non-stationary random process that can be modeled by piecewise stationary processes in the spatial domain. As such, it is modeled as a space-continuous Gaussian Mixture Model. Consequently, the model takes into account different regions of the scene, their edges, and their development along the spatial and disparity dimensions.

Applications presented include light field coding, depth estimation, edge detection, segmentation, and view interpolation. The representation is compact, which allows for very efficient compression yielding state-of-the-art coding results for low bit-rates. Furthermore, due to the statistical representation, a vast amount of information can be queried from the model even without having to analyze the pixel values. This allows for “blind” light field processing and classification

Index Terms— light field coding, depth estimation, light field representations, mixture-of-experts, mixture models

1. INTRODUCTION

Due to the curse of dimensionality, higher dimensional image modalities such as light fields (LFs) yield exponentially larger sets of samples than the lower dimensional modalities, e.g. still images or video. This results in new challenges for processing and coding this data. However, the increase of data allows for much more information to be extracted from the dataset. Our goal is to find a unified way of processing higher dimensional imagery that is statistically tractable.

Verhack et al. and Lange et al. proposed the *Steered Mixture-of-Experts* (SMoE) framework for images and video [1][2]. This work proposes the SMoE framework for light fields (LFs). The philosophy is deeply embedded in a Bayesian framework. The underlying assumption is that light

field rays are instantiations of a non-linear or non-stationary random process that can be modeled by piecewise stationary processes in the spatial domain. The model takes into account coherent regions of samples, their segmentation borders, spatial edges, and their development along any of the dimensions. Here, the process is modeled as a space-continuous *Gaussian Mixture Model* (GMM).

We illustrate some of the applications of the SMoE framework for LFs, i.e. edge detection, view inter- and extrapolation, depth estimation, and in particular: light field coding. An experimental evaluation is provided only for coding due to the limited scope of this work and the large current interest in the matter. No specific standard for light field images has been developed yet, it is the current goal of JPEG PLENO standardization initiative to do so [3]. Furthermore, SMoE has very beneficial features for coding, as the representation (1) is compact because of the sparse structure, (2) is analytical, continuous and closed-form, which makes SMoE resolution-independent, (3) allows for all other applications to be performed straight from the coded model, and (4) prioritizes dominant structures over small texture and noise.

Current coding solutions capture the projections of the underlying model at various discrete points in space and time and store these instantiations of the underlying function using an image format like JPEG or JPEG 2000, or a video format for image sequences such as H.264/AVC or H.265/HEVC. These formats exploit prediction and transform-based coding paradigms to represent the image or video data. Although content-adaptivity has proven to improve the compression efficiency, the fixed hybrid block-based structures have reached the limit of their potential because of their inflexible pre-engineered prediction and transform components. During the ICME Light Field 2016 Grand Challenge 4 out of 5 accepted submissions were H.265/HEVC-based [4]. The fifth submission was based on JEM, which is also prediction based using pseudo-sequences [5]. In the aforementioned techniques, whenever one view is to be decoded, a series of views that led to this requested view need to be decoded. The problem becomes even more cumbersome when a view is requested that was not explicitly captured (view-interpolation). Furthermore, the increase of image spatial resolution over the last

The research activities described in this paper were funded by IDLab (Ghent University - imec), Communication Systems Lab (Technische Universität Berlin), Flanders Innovation & Entrepreneurship (VLAIO), the Fund for Scientific Research Flanders (FWO Flanders), and the European Union.

decades gives more reason for sparse representations. Uniform pixel arrays do not tailor to the local spatial bandwidth requirements, which can vary heavily across an image.

We drastically depart from traditional block-based, transformation-based approaches. We propose a novel coding method for LFs. Our method aims at obtaining a compact continuous representation of the 4-D light field function. The encoder modeling and analysis task thus involves estimating the parameters of a GMM which models the joint probability density function. The coded file thus contains the parameters of the model. The expected value of the conditional density then serves as the regression function. The light field representation is reconstructed for any spatial position and viewing angle using a closed-form regression expression.

Depth map estimation is an important, well studied application from LFs. However, our approach is very different of all others. Most notably, our approach is able to reconstruct depth indicators directly from the coded model. We illustrate that our coded representation includes continuous depth information which could lead to new depth reconstruction techniques. Our approach has the similar goal of estimating continuous depth maps based on the slopes in epipolar plane images (EPIs) as in [6]. The slope of the line in the EPI plane are shown to be related to the depth of the corresponding point in space [7].

2. STEERED MIXTURE-OF-EXPERTS

2.1. Introduction

In the *Steered Mixture-of-Experts* (SMoE) framework, the underlying stochastic process of the amplitudes are modeled as a N -D multi-modal Mixture Model with K modes. This closed form continuous analytical model is estimated using e.g. the *Expectation-Maximization* (EM) algorithm and describes segments of samples by local N -D Gaussian steering kernels with global support. As such, each component in the Mixture-of-Experts steers along the direction of highest correlation. The expected value of the conditional density then serves as the regression function. Previous work has illustrated this idea for image and video luma coding, where the joint probability density function was modeled as respectively 3-D and 4-D Gaussian Mixture Models [1][2], and is here extended to color LFs.

Gaussian Mixture Models (GMM) offer elegant and relative easy descriptions for distributions and are frequently used to approximate multi-modal, multivariate distributions $p_{XY}(x, y)$. Given a GMM with K modes, one can derive a *Gaussian Mixture Regression* (GMR) as follows [8]. Assume training data $D = \{x^i, y^i\}_{i=1}^N$ with joint probability density

$$p_{XY}(x, y) = \sum_{j=1}^K \pi_j \mathcal{N}(\mu_j, R_j) \quad (1)$$

$$\text{and } \sum_{j=1}^K \pi_j = 1, \mu_j = \begin{bmatrix} \mu_{X_j} \\ \mu_{Y_j} \end{bmatrix}, R_j = \begin{bmatrix} R_{X_j X_j} & R_{X_j Y_j} \\ R_{Y_j X_j} & R_{Y_j Y_j} \end{bmatrix}$$

The parameters of this model are $\Theta = [\Theta_1, \Theta_2, \dots, \Theta_K]$, with $\Theta_j = (\pi_j, \mu_j, R_j)$, respectively being the population densities (or *priors*), centers, and covariances.

The conditional pdf $Y|X$ is given by [1]

$$p_Y(Y|X = x) = \sum_{j=1}^K w_j(x) \mathcal{N}(x; m_j(x), \sigma_j^2) \quad (2)$$

$$m_j(x) = \mu_{Y_j} + R_{Y_j X_j} R_{X_j X_j}^{-1} (x - \mu_{X_j}) \quad (3)$$

$$w_j(x) = \frac{\pi_j \mathcal{N}(x; \mu_{X_j}, R_{X_j X_j})}{\sum_{i=1}^K \pi_i \mathcal{N}(x; \mu_{X_i}, R_{X_i X_i})} \quad (4)$$

$$\sigma_j^2 = R_{Y_j Y_j} - R_{Y_j X_j} R_{X_j X_j}^{-1} R_{X_j Y_j} \quad (5)$$

Note that Eq. 4 corresponds to the *softmax* function frequently used in *artificial neural networks* and ensures the global support of the model. We can define the regression of the model to be equal to the expected value y given a sample location x through the conditional. From Eq. 2 and 4 follows the *regression function* $m(x)$:

$$m(x) = \sum_{j=1}^K w_j(x) m_j(x) \quad (6)$$

A signal at location x is estimated by the weighted sum over all K mixture components (Eq. 6). Every mode in the mixture model is treated as an expert and the experts collaborate towards the definition of the regression function. Note that the reconstruction is *smoothed piecewise linear* [1].

2.2. Light field reconstruction

We adopt the proposed two-plane parametrization $L(x, y, s, t)$ from Levoy et al. [11]. However for consistency with other SMoE papers, we will here use the variables $LF(a_1, a_2, x_1, x_2) = (Y, Cb, Cr)$, with a_1, a_2 being the directional dimensions, x_1, x_2 the spatial dimensions. $LF(a_1, a_2, x_1, x_2)$ will as such give the pixel amplitudes for horizontal parallax at disparity a_1 , vertical disparity a_2 , row x_1 , and column x_2 . This is the datastructure that is yielded by the Light Field Toolbox v0.4 [12]. Consequently, our GMM is 7-D, X being 4-D and Y being 3-D in Eq. 1.

Note that the soft-windows $w_j(x)$ (Eq. 4), describe a 4-D volume per component in which the component acts as the expert. The response of this expert $m_j(x)$ (Eq. 3) describes for each color channel a hyperplane in the 4-D space. This is visualized in Fig. 1. Fig. 1a shows the epipolar planes (EPI) for the original LF. The red lines indicate where 4-D space is “cut”. Fig. 1b shows a low order GMM fit onto the data, note that our components are 4-D Gaussians. Fig. 1c and 1d visualize the 3-D projection onto (a_1, x_1, x_2) . Fig. 1e show the segmentation, which is nothing more than the hard-decision of our soft-windows $w_j(x)$. It is clear that our windows soft-partition the entire 4-D space, thus yielding global support. Finally, the reconstruction based on Eq. 6 is illustrated in 1f.

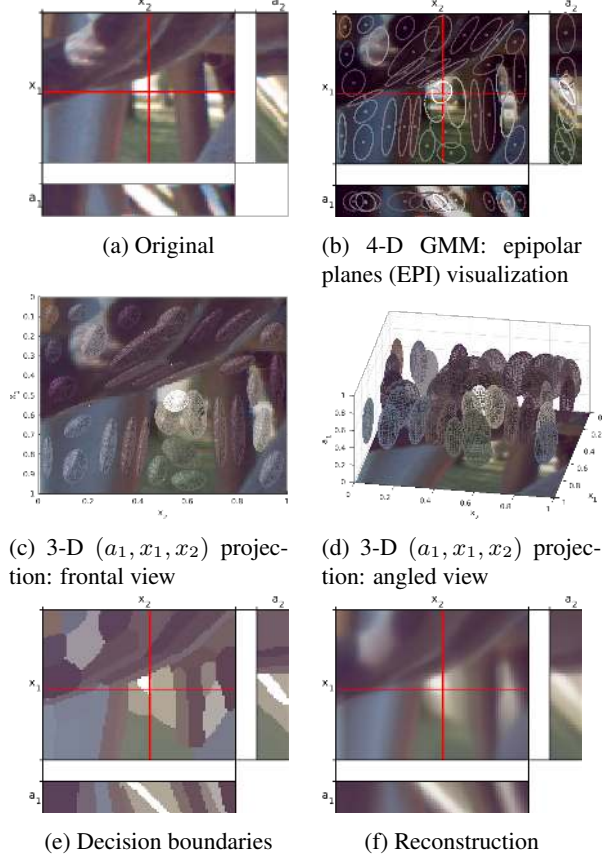


Fig. 1: *101 Bikes* [4][9] crop modeling with a very low amount of kernels for visualization ($K=35$).

Fig. 2b shows the reconstructed (7,7)-view from the *101 Bikes* LF shown in Fig. 2a [4]. The modeling is detailed in Sec. 3. Note how the speckle rust turns into smudges in the reconstruction, which yields a visually pleasing quality decay. This is however heavily penalized when objective metrics are used. Note that, the reconstruction is a bit blurred due to the relatively low number of components ($K = 8960$) compared to 41.483.904 original pixels in the lenslet image. Thus resulting in 4.630 pixels for one component on average, i.e. each 4-D soft-window spans 4.630 samples on average.

Important to note is that our method is able to reconstruct views that were not captured. Our model has a continuous representation, as such any view in the domain can be readily be reconstructed. Limited extrapolation is also possible. Fig. 3a, shows that the LF datastructure (obtained through the aforementioned LF Toolbox) results in black views in the corner directional views. Our method is able to estimate these views with remarkable consistency by excluding the black views during training. The effect is clearly visible by the position of the red square on the background (Fig. 3b).

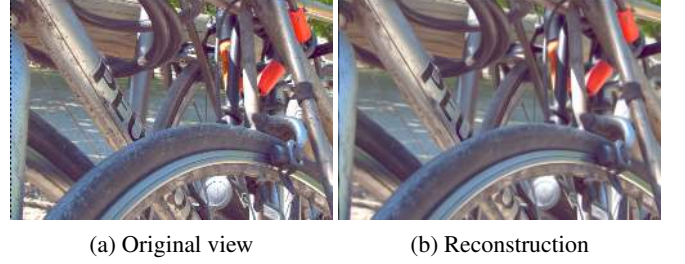


Fig. 2: *101 Bikes* [4][9] light field example ($K=8960$), showing a central view with $(a_1, a_2) = (7, 7)$. Mean $\text{PSNR}_{\text{YCbCr}}$: 30.71 dB, mean SSIM_Y : 0.86 (objective evaluation as in [4])

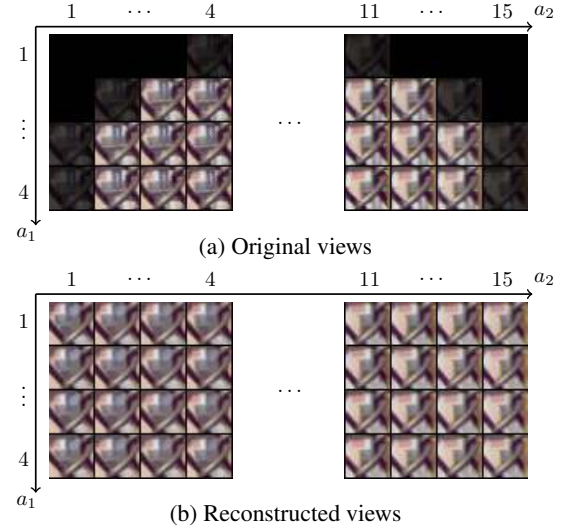


Fig. 3: View reconstruction (26x27 spatial crop from *102*)

2.3. Depth estimation

The slopes of the lines in the EPI plane are related to the depth of the corresponding points in the 3-D space [13]. For each pixel, the slope in the EPI planes is determined by the weighted sum of the spatio-directional covariances. This results in two depth estimations. For each component we defined the horizontal α_H as follows:

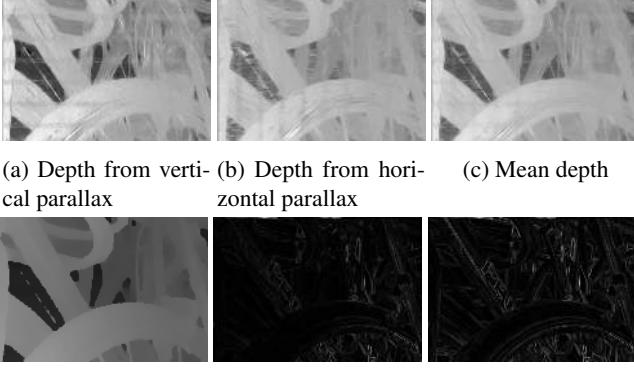
$$\alpha_H = \text{atan2}(e_H) \quad (7)$$

with e_H being the largest eigenvector of the 2-D covariance matrix of a_1 and d_1 . Analogously for α_V with the covariance matrix of a_2 and d_2 .

To obtain a continuous angle value for a pixel i , the same weighted sum can be used as for the regression:

$$\alpha_{i,H} = \sum_{j=1}^K w_j(x_i) \alpha_H^j \quad (8)$$

Results are shown in Fig. 4. It is clear that this simple approach is able to perform an interesting depth estimation.



(a) Depth from vertical parallax (b) Depth from horizontal parallax (c) Mean depth
(d) Depth map from Lytro software [10] (e) 2-D edge detection (f) 4-D edge detection

Fig. 4: Depth estimation and edge detection through gradient strength from model in Fig. 2

The estimation is noisy, but is continuous and detailed, compared to the depth map in Fig. 4d provided by the Lytro software [10]. Note the very narrow brake cable under the “Peugeot” beam. The vertical and horizontal block artifacts are caused by a block-wise modeling approach, detailed in Section 3 (blocksize=128).

2.4. 4-D edge detection and other descriptors

In [1], an image edge detection method was proposed based on SMOE. Each component describes an edge which is described by the norm of $R_{Y_j, X_j} / R_{X_j, X_j}$. The same can be done for LF in 4-D. Consequently, not only are the spatial gradients taken into account, but also the variation of the luma along the directional dimensions. Fig. 4 shows an example comparison of using only the spatial dimensions for edge detection (x_1 and x_2), and the situation where the full covariance is used (a_1, a_2, x_1 , and x_2). 4-D edge detection results in a much more detailed result as information is used from multiple viewing angles.

Note that the other 2-D image features presented in [1], are readily extensible to 4-D. These methods (4-D segmentation, flow, ...) are out of the scope of this work.

3. MODEL CODING

3.1. Introduction

Our coding strategy follows the same principle as the 2-D image case [1], with some adjustments. In contrary to the work of Verhack et al. [1], the priors π_j are not estimated at decoder side. Instead, the models were trained by constraining the priors to be $1/K$. Furthermore, during these experiments the chroma slope components $R_{X_j, Y_{Cb, Cr, j}}$ are discarded, as the human visual system is not very sensitive to those changes. Note that both the modeling and the coefficient quantization contribute to the approximation error.

3.2. Local Modeling

The Expectation-Maximization (EM) algorithm is used to estimate the parameters $\Phi_j = (\pi_j, \mu_j, R_j)$ for each component j [14]. The modeling is performed on spatially restricted overlapping blocks ranging over the full directional dimensions. This is in contrast to the non-overlapping blocks in [1]. The overlap mitigates the abrupt changes around block-edges previously visible, which are due to data truncation. Furthermore, in contrary to [1], all blocks in this work received the same budget of components. As such, we limit the amount of free parameters in this early work and the prior can be assumed to be uniform.

3.3. Window $R_{X_j X_j}$ quantization

In contrary to the work in [1], we employ a vector quantization-like method for coding the window covariance $R_{X_j X_j}$. We propose an EM-like algorithm based on the Kullback-Leibler (KL) divergence. As such, the probability densities are compared, which are more informative than the covariance parameters. We normalize all $R_{X_j X_j}$ by $|R_{X_j X_j}|^{(1/d)}$. In the case of $R_{X_j X_j}$ for LFs, d equals 4. As such, the constructed codebook contains normalized shapes with determinants of one. The coding of the magnitude of the shape, i.e. $|R_{X_j X_j}|^{(1/d)}$ is discussed in the next subsection.

The KL-divergence for multivariate Gaussians $P \sim \mathcal{N}(\mu_P, R_P)$ and $Q \sim \mathcal{N}(\mu_Q, R_Q)$ is given by

$$D_{KL}(P \parallel Q) = \frac{1}{2} \left[\log \left(\frac{|R_P|}{|R_Q|} \right) - d + \text{tr}(R_Q^{-1} R_P) \right] + \frac{1}{2} \left[(\mu_Q - \mu_P)^T R_Q^{-1} (\mu_Q - \mu_P) \right] \quad (9)$$

As our data is normalized, $|R_P|$ and $|R_Q|$ equal one. Furthermore, the windows are assumed to be centered on the origin, i.e. μ_P and μ_Q are zero. In order to obtain a symmetric similarity measure, we define our distance as

$$d(P, Q) = \frac{D_{KL}(P \parallel Q) + D_{KL}(Q \parallel P)}{2} = \frac{1}{4} \left(-2d + \text{tr}(R_Q^{-1} R_P) + \text{tr}(R_P^{-1} R_Q) \right)$$

Covariances are clustered around a centroid using $d(P, Q)$ and at each iteration the new centroid covariance C_l is calculated as the mean covariance of the members of the cluster l and renormalized. Fig. 5 illustrates the algorithm on a 2-D dataset.

This codebook was trained at encoder side, and transformed to ensure robustness. As each C_l is semi-positive definite, C_l can be decomposed using Cholesky: $C_l = A^T A$. A is vectorized and each coefficient is coded analogously to the slopes $R_{X_j Y_j}$ (see following subsection). At decoder side, the multiplication $A^T A$ ensures the reconstructed covariance to be semi-positive definite again.

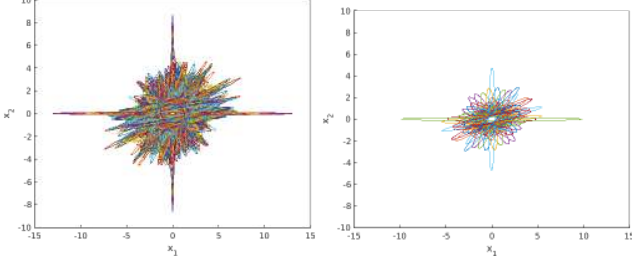


Fig. 5: Codebook of size 64 (right) made from a set of 8960 2-D normalized covariances (left)

3.4. Center μ and slope $R_{X_j Y_j}$ quantization and arithmetic coding

The centers $\mu = [\mu_X, \mu_Y]$ are difference coded by defining a path that comprises every component in a greedy fashion. Start with the component j closest to $(0, 0)$. Find component $k, (k \neq j)$, so that $|\mu_j - \mu_k|$ is minimal.

The 7-D differenced centers $\mu_j = [\mu_j^1, \dots, \mu_j^7]$, the 4 dimensions in $R_{X_j Y_j}$ and the shape magnitude $|R_{X_j Y_j}|^{(1/d)}$ are assumed to be Laplacian distributed. These values are concatenated in one 12-D vector s and are further normalized as follows

$$\tilde{s}_j^i = \frac{s_j^i - E[s^i]}{c^i \sigma_{s^i}} \quad (10)$$

with $c^i \geq 1$ being the ratio determining how much more subsampled the coefficient i needs to be compared to the location. We set $c^3 = c^4 = 1$ as the baseline, i.e. c^i with $i \notin \{3, 4\}$ determines how much less important coefficient i is compared to the spatial location center (μ_j^3, μ_j^4) . Hereby we assume that the precision of the location always will be the highest compared to the other coefficients. Consequently, the distribution of the coefficient i with $c^i > 1$ is squeezed together. Next, quantization is performed uniformly based on the limits of s . As such, we are able to combine different quantization steps for each coefficient, while still using a single arithmetic coder. The same Laplacian adaptive arithmetic coder is employed as in [15].

4. CODING EXPERIMENTS

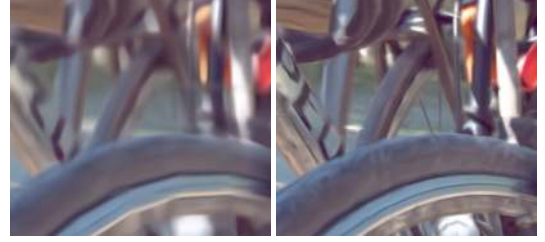
4.1. Experiment setup

In this work, our quantitative analysis is limited to the coding performance and exact view reconstruction. We followed the same set up as the ICME 2016 Grand Challenge [4], with the difference that SMoE operates directly on the 4-D LF stack, and not on the lenslet image.

The parameters were found using random search due to the large amount of parameters: blocksize, kernels per block K_i , quantization steps, book size, ... The quantization step ranged $[7, 8, 10, 12]$, ratios $c_i = [1, 2, 8, 16]$, book sizes = $[2^6, 2^8, 2^9]$, blocksize = $[32, 64]$, and $K_i = [8, 16, 32, 64, 96, 128]$. To lower the computational demand,



(a) JPEG on lenslet (0.14, 26.47, 0.70) [4] (b) JPEG-2K view encoding (0.05, 22.30, 0.66)



(c) SMoE (0.002, 22.97, 0.66) (d) SMoE (0.01, 26.43, 0.80)



(e) SMoE (0.028, 27.75, 0.87) (f) Original

Fig. 6: Visual comparison of one view $(a_1, a_2) = (7, 7)$ at different bitrates. Bitrates are calculated as the LF filesize in bits divided by the number of pixels in the lenslet image as in [4]. Except for (b): the filesize of the J2K coded file was used, with the number of pixels of one view. Metrics are in the format (bpp, PSNR (dB), SSIM).

and due to the extreme redundancy in a single LF, we trained our models on a quasi-random subset of 25% of the samples using the Sobol sequence.

4.2. Coding performance

Fig. 7 shows the rate-distortion (RD) curves on a log scale for three LFs: *I01, I02*, and *I10*, optimized to PSNR. Note that the bitrates evaluated in [4] only start at 0.1 bpp, whereas our range starts at 0.001 bpp. We did not have access to the decoded images in [4], so here we can only compare the objective metrics. Note that due to the approximative nature of our method, near-lossless quality is hard to obtain because of the intrinsic filtering and small geometrical distortions. Especially due to the fact that we only train on a percentage of the original amplitudes. As such, the proponents in [4] yield higher PSNR (> 32 dB for all three images) at 0.1 bpp, than

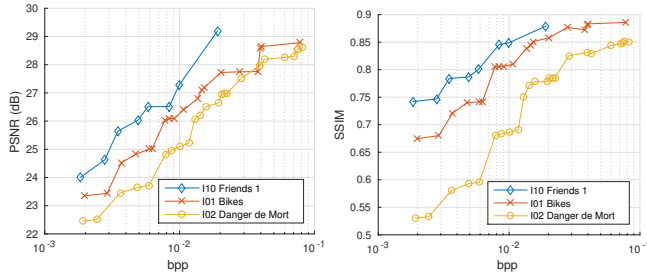


Fig. 7: Rate-distortion curves for I01, I02, and I10 [4].

SMoE at our highest RD-point.

A visual comparison with the JPEG-baseline at 0.14 bpp is provided in Fig. 6 [4]. Reconstruction from lenslets are very sensitive towards artifacts in the fine details of the lenslet image. As such, the JPEG lenslet coding yields sharp but corrupted views as in Fig. 6a. Fig. 6b shows JPEG-2000 on a single view (instead of coding the lenslet image). This also provides no bitrates as low as ours. SMoE coded views are shown for low, mid and higher quality, at very low bitrates. It is clear that even at lower quality, the artifacts introduced by SMoE are less outspoken than for JPEG lenslet coding.

5. CONCLUSIONS

The SMoE approach for LF representation and coding introduced in this paper drastically departs from traditional techniques. Long-range spatial and angular correlations are explored in a unified Bayesian approach. Large 4-D volumes of pixel data are harvested by optimizing component locations and steering parameters. The resulting kernels are edge-aware and steer along spatial and angular directions. The kernels act as experts and all collaborate towards the reconstruction of the LF images.

A steered kernel network provides sophisticated disparity, depth, and edge clues. Since the approach derives a closed-form spatio-angular continuous regression equation for pixel amplitudes, intermediate views as well as extrapolated views are readily available at any spatial and angular resolution, scale and location. Results presented for compression of LFs indicate that the SMoE approach can be highly competitive at low rates.

6. REFERENCES

- [1] R. Verhack, T. Sikora, L. Lange, G. Van Wallendael, and P. Lambert, "A universal image coding approach using sparse steered Mixture-of-Experts regression," in *2016 IEEE International Conference on Image Processing (ICIP)*. sep 2016, pp. 2142–2146, IEEE.
- [2] L. Lange, R. Verhack, and T. Sikora, "Video Representation and Coding Using a Sparse Steered Mixture-of-Experts Network," in *Picture Coding Symposium (PCS)*, 2016.
- [3] P. Schelkens, "JPEG PLENO-Scope, use cases and requirements Ver. 1.3," *ISO/IEC JTC*, vol. 1.
- [4] I. Viola, M. Rerabek, T. Bruylants, P. Schelkens, F. Pereira, and T. Ebrahimi, "Objective and subjective evaluation of light field image compression algorithms," in *32nd Picture Coding Symposium*, 2016, number EPFL-CONF-221601.
- [5] D. Liu, L. Wang, L. 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)*. jul 2016, pp. 1–4, IEEE.
- [6] Jianqiao Li, Minlong Lu, and Ze-Nian Li, "Continuous Depth Map Reconstruction From Light Fields," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3257–3265, nov 2015.
- [7] S. Wanner and B. Goldluecke, "Globally consistent depth labeling of 4D light fields," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. jun 2012, pp. 41–48, IEEE.
- [8] H. Sung, *Gaussian Mixture Regression and Classification*, Ph.D. thesis, Rice University, 2004.
- [9] M. Rerabek and T. Ebrahimi, "New Light Field Image Dataset," in *8th International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, number EPFL-CONF-218363.
- [10] Lytro Inc., "Lytro Desktop Application v5.0.1," 2015.
- [11] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques - SIGGRAPH '96*, New York, New York, USA, 1996, pp. 31–42, ACM Press.
- [12] D. G. Dansereau, "Light Field Toolbox for Matlab," 2015.
- [13] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 7–55, 1987.
- [14] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.
- [15] R. Verhack, L. Lange, P. Lambert, R. Van de Walle, and T. Sikora, "Lossless image compression based on Kernel Least Mean Squares," in *2015 Picture Coding Symposium (PCS)*. may 2015, pp. 189–193, IEEE.