Steered Mixture-of-Experts for Light Field Images and Video: Representation and Coding

Ruben Verhack, Thomas Sikora, Glenn Van Wallendael, and Peter Lambert

***Abstract*—Research in light field (LF) processing has heavily increased over the last decade. This is largely driven by the desire to achieve the same level of immersion and navigational freedom for camera-captured scenes as it is currently available for CGI content. Standardization organizations such as MPEG and JPEG continue to follow conventional coding paradigms in which viewpoints are discretely represented on 2-D regular grids. These grids are then further decorrelated through hybrid DPCM/transform techniques. However, these 2-D regular grids are less suited for high-dimensional data, such as LFs. We propose a novel coding framework for higher-dimensional image modalities, called Steered Mixture-of-Experts (SMoE). Coherent areas in the higher-dimensional space are represented by single higher-dimensional entities, called kernels. These kernels hold spatially localized information about light rays at any angle arriving at a certain region. The global model consists thus of a set of kernels which define a continuous approximation of the underlying plenoptic function. We introduce the theory of SMoE and illustrate its application for 2-D images, 4-D LF images, and 5-D LF video. We also propose an efficient coding strategy to convert the model parameters into a bitstream. Even without provisions for high-frequency information, the proposed method performs comparable to the state of the art for low-to-mid range bitrates with respect to subjective visual quality of 4-D LF images. In case of 5-D LF video, we observe superior decorrelation and coding performance with coding gains of a factor of 4x in bitrate for the same quality. At least equally important is the fact that our method inherently has desired functionality for LF rendering which is lacking in other state-of-the-art techniques: (1) full zero-delay random access, (2) light-weight pixel-parallel view reconstruction, and (3) intrinsic view interpolation and super-resolution.**

***Index Terms*—Mixture of experts, light fields, mixture models, sparse representation, bayesian modeling.**

Manuscript received May 30, 2018; revised February 13, 2019 and May 7, 2019; accepted July 20, 2019. Date of publication August 1, 2019; date of current version February 21, 2020. This work was supported in part by IDLab (Ghent University - imec), in part by Communication Systems Group (Technische Universität Berlin), in part by Flanders Innovation & Entrepreneurship (VLAIO), in part by the Fund for Scientific Research Flanders (FWO Flanders), and in part by the European Union. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mea Wang. *(Corresponding author: Ruben Verhack.)*

R. Verhack is with the IDLab, Ghent University –imec, B-9052 Ghent, Belgium, and also with the Technische Universität Berlin –Communication Systems Group, 10623 Berlin, Germany (e-mail: ruben.verhack@ugent.be).

T. Sikora is with the Technische Universität Berlin –Communication Systems Group, 10623 Berlin, Germany (e-mail: sikora@nue.tu-berlin.de).

G. Van Wallendael and P. Lambert are with the IDLab, Ghent University –imec, B-9052 Ghent, Belgium (e-mail: glenn.vanwallendael@ugent.be; peter.lambert@ugent.be).

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

Digital Object Identifier 10.1109/TMM.2019.2932614

I. Introduction

VIRTUAL reality (VR) for camera-captured scenes is fundamentally different and more complex compared to VR consumption of computer-generated (CG) scenes (e.g. as in gaming). The problem is much more challenging due to the lack of geometrical knowledge. 360° video is inadequate at delivering a full VR experience as it does not allow for important visual clues (e.g., parallax) and does not deliver positional freedom to the user. As such, there is a large interest in overcoming these limitations and allowing the same level of autonomy for the viewer.

Two general strategies exist for allowing true immersive experiences of camera-captured scenes. The first strategy consists of reverse-engineering a 3-D model. Such methods rely on image-based 3-D modeling techniques that require a large number of fixed cameras. After the 3-D model construction, a texture is estimated based on the input images. All these approximation steps are inherently lossy, time-consuming and often require manual intervention (e.g., to address temporal consistency of 3-D models). Furthermore, they struggle with non-rigid objects such as smoke, fire, water, and transparent surfaces. The decoder then consists of a traditional 3-D renderer, in which the decoding speed heavily depends on the scene's complexity. The second strategy relies on known hybrid transform/difference-coding techniques commonly used for video. Following this philosophy, scenes are represented by coding a minimal set of 2-D images, and reconstructing the missing ones by view synthesis. These methods provide excellent coding performance for video and for problems that can be translated to video. However, classical video coding is becoming less translatable (and thus less applicable) to VR, especially with future extensive user autonomy. Firstly, the number of possible views grows exponentially with the level of user's autonomy. Secondly, the time-order in video is adequately exploited by motion-compensation, however, in VR, it is difficult to exploit the order of the frames as the end-user defines the order at playback. Thirdly, the decoder complexity grows when less views are transmitted due to a more complex view synthesis process. Fourthly, the reconstruction is not truly pixel-level parallel due to the intra-coding techniques. Finally, these systems do not cope well with irregularly-sampled data and heterogeneous camera setups in scene acquisition systems.

In this paper, we propose a novel third strategy which we believe is a more adequate fit for higher-dimensional image data. Our *Steered Mixture-of-Experts* (SMoE) methodology models the light information as higher-dimensional data. More specifically, we sparsely approximate the underlying pixel-generating

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

plenoptic function using inherently higher-dimensional atoms called 'kernels'. These kernels allow for simultaneous harvesting of pixel color correlation in various directions: e.g. time, pixel position, camera position. In essence, the observed 2-D views in a VR scene at each position and gaze orientation are 2-D projections of higher-dimensional light data. Our belief is that we should accept the fact that light data is intrinsically of high dimensionality, and we should not reduce the problem to 2-D problems. Our aim is to embrace the high-dimensional nature and to develop a tailored technique. Furthermore, this data is likely to be irregularly positioned due to novel complex acquisition systems [1]. We argue that 2-D regular sampling grids are thus not optimal representations for storing high-dimensional data, such as light fields (LFs). The light information in a physical space is mathematically defined by the 5-D plenoptic function (discarding time): $I(X, Y, Z, \alpha, \gamma)$ yielding the color and intensity of the incoming ray at location (X, Y, Z) and angles (α, γ) [2]. However, under the assumption of "open space" (no occluding objects), the 5-D plenoptic function can be reduced to the 4-D light field [3].

Our proposed representation is sparse and consequently compact, which is well suited for compression. Furthermore, our representation leverages desired functionality for VR consumption for the following reasons. Firstly, information about the light incoming at any orientation at a certain point is locally available in our model and can be decoded independently. This yields potential for granular random access. Secondly, our representation is continuous and is thus unaware of resolution at encoding and decoding side. The modeling takes any irregularly-sampled data and reconstructing a view at arbitrary position boils down to merely sampling our representation at a desired output resolution, independent of the other reconstructed pixels. Consequently, pixel-parallel rendering is made possible [4].

SMoE was previously proposed for images and video, and has the extraordinary property of scaling towards images of arbitrary dimensionality [5], [6]. In this paper, we focus on LF images and video, and more specifically the reconstruction performance with coded model parameters as a LF compression tool. Other applications of this representation are super-resolution, denoising, segmentation, etc. However, these are not the focus of this paper. An initial short paper on SMoE for 4-D LFs has been published before [7] and results were used in a comparative study [8], but suffered from unstable modeling which is examined and fixed in this work. The novelties of this paper are: (1) an in-depth presentation of SMoE (even though some parts have been presented scattered over multiple short papers [5]–[7], no sufficiently complete presentation was present to date); (2) faster and more robust modeling using minibatches (x100 speed-up) and a detailed evaluation; (3) a better quantization of the covariance matrices; and (4) novel objective and subjective experimental validation on the coding performance for 4-D LF images and objective coding results for 5-D LF video.

The paper is structured as follows. Section II discusses prior work, standardization efforts, and related work. Next, Section III introduces the general theory in SMoE. Section IV discusses the application and properties of SMoE applied to 2-D images, LF images and LF video. We propose a coding method based on the SMoE representation in Section V and evaluate the proposed robust modeling and coding in Section VI. Finally, we present our conclusions and future work in Section VII.

II. Related Work

The standardization organization *Moving Pictures Experts Group* (MPEG) is considering *Depth-Image-Based Rendering* (DIBR) for their efforts in standardizing a codec that allows for broad Six Degrees-of-Freedom (6-DoF, i.e. full translational and gaze direction freedom) which is targeted for 2021 [9], [10]. This will likely build further on the 3-D extension of *High Efficiency Video Coder* (HEVC), which allows for the usage of geometrical side-information for free-viewpoint navigation [11]. MPEG's vision consists of two phases at the encoder side: (1) identifying a minimal set of representative views and (2) compressing these views [9]. The receiving side then performs some view synthesis. However, the acquisition of this geometrical side information suffers from the same inherent problems as 3-D reverse-engineering techniques. Finally, even methods for coding holographic data by using HEVC have been proposed [12].

Light field imaging allows for photorealistic reproduction of a real scene without geometrical information, however, it limits the user's autonomy, i.e. the user can not step "into the scene" within the objects, but remains an outside viewer [1], [3]. The extent of translatory movement depends on the width of the camera array and defines the viewable region on the camera plane. The resolution within this region is defined by the length of the baseline, i.e. the inter-camera distance [1]. The shortest possible baselines are typically captured using a microlens array on top of a single image sensor, dubbed *lenslet* cameras, e.g. Lytro cameras [13]. In case of the specific lenslet sensor hardware, intra-prediction can be applied directly on the lenslet sensor image. Examples include self-similarity [14] and local linear embedding [15] intra-prediction methods, both embedded into HEVC. However, these methods are only applicable for these specific lenslet hardware architectures.

A more hardware-agnostic approach handles the light field more generally as a 2-D matrix of 2-D camera views. Such coding schemes often rely on video coding techniques by forming a pseudo video-sequence of the captured views that serves as input for HEVC [16]. This is commonly used as an anchor in light field coding [17], and is also used as such in this article. A hierarchical reference structure was proposed for inter-coding of the pseudo video-sequence [18]. Furthermore, a coding method that allows for Field-of-View scalability was recently proposed using HEVC as a base layer combined with an exemplar-based inter-layer prediction [19]. A method based on the multiview extension of HEVC (MV-HEVC) has also been proposed [20]. A similar technique to MPEG's 6-DoF vision is present in the work on lenslet light fields using structural key views [21]. Ideas of compressed sensing are incorporated in order to achieve minimal parameters to define the whole light field. An effective, although elaborate view synthesis for wide-baseline camera arrays was recently introduced in [22], and is considered to be the state of the art at the time of writing.

Recently, JPEG has started standardization activities for coding methods targeting light fields. This is as part of their larger

ambition of JPEG-Pleno, which is aimed to arrive at a single unifying format for point-clouds, LFs and holographic image data [23]. One promising method under consideration is a 4-D DCT-based codec [24]. This method achieves very competitive results on lenslet LFs using a conceptually-simple design. However, it remains a dense representation that requires regularly sampled data, and also requires as many coefficients as there are pixels in all views. Furthermore, the efficiency on wide-baseline light fields is not ensured as larger shifts in views introduce discontinuities along camera dimensions and discontinuities are usually not well represented by a DCT.

Our work is inspired by the works of Mumford-Shah, Prandoni & Vetterli, and Takeda [25]–[27]. The Mumford-Shah variational model shows that natural images are characterized by having regions that behave smooth but are separated by discontinuities (edges) [25]. This allows us to assume images have a piecewise stationary nature. Prandoni & Vetterli's theoretical and experimental work on the approximation and compression of piecewise smooth functions showed that for such functions, a sparse coding scheme is much more efficient than fixed grids [26]. Takeda introduced *Steered Kernel Regression* (SKR) for image processing in which kernels were steered along image features in order to perform denoising, super-resolution, etc [27]. In our proposed method, we combine these ideas to represent the coherent regions in image modalities by a sparse set of kernels. The introduced SMoE approach borrows many concepts from non-linear regression techniques in the machine learning world in which kernel approaches are well-known, i.e. *Radial Basis Function Networks* (RBF) [28] and non-linear *Support Vector Regression* (SVR) [29].

Early work already employed SKR for image coding as a reconstruction of the full image that was stored heavily subsampled [30]. In contrast, our proposed approach has the stark difference that the kernel parameters are stored instead of pixel data. Furthermore, these kernels can be irregularly positioned. One limitation of SKR was the limited support by kernels, i.e. there is no guarantee that all pixels are covered by a kernel. In SMoE, each expert function has global support and is weighted to ensure that each pixel has support. One remarkable property of SMoE is that it is applicable to data of any dimensionality, which lifts the limitation that Prandoni & Vetterli had in their technique that did not scale to 2-D.

Other related work also involves sparse light field coding in order to reveal scene structure [31], [32]. As in SMoE, the central idea is to represent a signal as a linear combination of some core atoms. However, SMoE's atoms are Gaussian kernels that have a certain position in the coordinate space, whereas the elements in [32] belong to a patch dictionary. Similar to our kernel representation idea is the work on identifying coherent 4-D atoms in light fields, i.e. "superrays" for efficient light field processing [33]. SMoE provides a continuous representation of the 4-D light field. Similarly, the excellent work on shearlets also approximates a continuous camera plane based on a limited set of views for light field reconstruction [34]. Finally, higher-dimensional image modeling is also typically found in medical image processing in e.g. 4-D perfusion reconstruction [35].

III. Steered Mixture-of-Experts (SMoE)

A. Introduction

In SMoE, we approximate image modalities, and in general signals, by modeling them as a set of coherent kernels. We define a coordinate space $\mathbb{R}^p$ and a color space $\mathbb{R}^q$. For images, video, LF images, and LF video, the dimensionality p is respectively 2, 3, 4, and 5. For monochrome images, q is 1, and for color images q is typically 3. The underlying assumption is that image pixels are instantiations of a non-linear or non-stationary random process that can be modeled by spatially-piecewise stationary stochastic processes, very similar to the Mumford-Shah variational model for 2-D images [25]. As such, the model takes into account different regions of the image and their segmentation borders. Furthermore, in light fields, the epipolar-plane images consist of diagonally-structured lines, and in video, motion can be approximated by line segments as is done in motion compensation in HEVC. Therefore, we target a piecewise approximation of image modalities.

The goal is to divide the coordinate space X into stationary regions, and to find local regressors ($f : \mathbb{R}^p \mapsto \mathbb{R}^q$) that locally approximate this stationary region well. This is the general *Mixture-of-Experts* (MoE) strategy, well known in the machine learning field. However, SMoE is based on a mixture model (or "alternative") version of the MoE approach [36]. In this version, both segmentation and local regressors (the *experts*) are derived from the modes of a mixture model. This mixture model models the joint probability distribution of the random vectors $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}^q$ in respectively the coordinate space and color space.

In this paper, we limit ourselves to the *Gaussian Mixture Model* (GMM) case. One Gaussian kernel in the model defines a linear regressor through the conditional $Y|X$, and all kernels combined define a segmentation of the coordinate space. The model thus only consists of a set of Gaussian kernels which are defined by their centers and covariances. The reason for choosing GMMs is that they offer elegant mathematics and limited parametrization. Furthermore, the MoE based on GMMs results in smoothed piecewise approximations, which fit natural image modalities quite well, as mentioned above. However, the linear nature might fail to capture high spatial frequencies such as noise and fine texture. Therefore, we do not exclude future models with more expressive regressors. The parameters of these kernels are found using likelihood optimization. Consequently, the kernels harvest correlation over all dimensions and steer along the dimensions with highest correlation. As such they align with e.g. edges (in spatial dimensions) and temporal flow (in the time dimension) in the case of video [5], [6].

The next subsections provide firstly, the theory of Mixture-of-Experts based on GMMs. Secondly, we discuss the improved training method used in this work and, finally, we illustrate the concept of SMoE on a 1-D signal.

B. Mixture-of-Experts Based on GMMs

In general, the goal of regression is to optimally predict a realization of a random vector $Y \in \mathbb{R}^q$, based on a known random vector $X \in \mathbb{R}^p$. MoE regression follows a tree structure

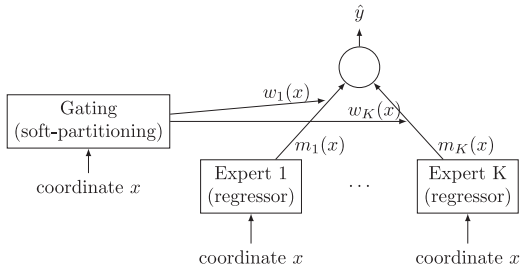


Fig. 1. Illustration of a Mixture-of-Experts with one layer for regression. The gating function soft-partitions the input space in regions where particular experts (in this case regressors) are the most influential.

as illustrated in Fig. 1. Given K experts with gate parameters $\Theta_g = \{\theta_{g,j}\}_{j=1}^K$, expert parameters $\Theta_e = \{\theta_{e,j}\}_{j=1}^K$, the total probability of observing $\boldsymbol{y}$, given an input vector $\boldsymbol{x}$, can be written in terms of the experts, as

$$p_Y(\boldsymbol{y}|X=\boldsymbol{x},\Theta) = \sum_{j=1}^{K} \underbrace{p_X(j|\boldsymbol{x},\theta_{g,j})}_{\text{gate access}} \underbrace{p_Y(\boldsymbol{y}|\boldsymbol{x},\theta_{e,j})}_{\text{expert posterior}}. \quad (1)$$

Due to the modular structure, the gates can be placed in a tree-structure forming hierarchical MoEs (HME) [36]. The original MoE approach and modeling differentiated between the model parameters for the gates Θ_g and for the experts Θ_e, and relied on iteratively recursive least mean squares (IRLS) for estimating the expert parameters.

In the following, we will elaborate on the "alternative" MoE definition which is deeply rooted in a Bayesian framework based on GMMs [36]. This method has the advantage that both the gates Θ_g and the experts Θ_e are simultaneously defined by the Gaussian components of the mixture model. Thus, the estimation of the parameters for gates and experts are optimized simultaneously and IRLS is not needed.

Consider a mixture of distributions of which the joint probability is given by

$$p_{XY}(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{K} \pi_j \phi_j(\boldsymbol{x},\boldsymbol{y};\theta_j), \quad (2)$$

with π_j being the prior for distribution ϕ_j.

Regressing the mixture model is equal to finding a measure of central tendency. For example, the expectation or maximum-a-posteriori of Y given X of the mixture model, e.g. the mean, median and mode of the marginal $p_Y(Y|X=\boldsymbol{x})$. Note that the mean is the easiest to compute, and does not rely on the variance of $p_Y(Y|X=\boldsymbol{x})$. As such, less information needs to be transmitted in the case of coding. We will focus on the expected value of the conditional: $\mathbf{E}[Y|X=\boldsymbol{x}]$.

GMMs offer elegant and relatively-easy descriptions for distributions and are frequently used to approximate multi-modal, multivariate distributions $p_{XY}(\boldsymbol{x},\boldsymbol{y})$. Given a GMM, one can derive a regression as follows [37], [38]. Assume data $D = \{\boldsymbol{x}^i, \boldsymbol{y}^i\}_{i=1}^N$ with joint probability density:

$$p_{XY}(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\boldsymbol{x},\boldsymbol{y};\boldsymbol{\mu}_j, R_j) \quad (3)$$

and $\sum_{j=1}^K \pi_j = 1, \boldsymbol{\mu}_j = [\begin{smallmatrix}\boldsymbol{\mu}_{X,j}\\ \boldsymbol{\mu}_{Y,j}\end{smallmatrix}], R_j = [\begin{smallmatrix}R_{XX,j} & R_{XY,j}\\ R_{YX,j} & R_{YY,j}\end{smallmatrix}]$.

The parameters of this model are $\Theta = [\theta_1, \ldots, \theta_K]$, with $\theta_j = (\pi_j, \boldsymbol{\mu}_j, R_j)$, respectively being the priors, centers, and covariances. A normal *probability density function* (pdf) of dimension $p+q$ can be factorized as

$$\mathcal{N}_{p+q}\left(\begin{bmatrix}\boldsymbol{\mu}_X\\ \boldsymbol{\mu}_Y\end{bmatrix}, \sigma^2\right) = \mathcal{N}_q(\boldsymbol{\mu}_{Y|X}, R_{Y|X})\mathcal{N}_p(\boldsymbol{\mu}_X, R_{XX}),$$

where $R_{Y|X}$ is the *Schur* complement:

$$R_{Y|X} = R_{YY} - R_{YX}R_{XX}^{-1}R_{XY}. \quad (4)$$

Accordingly the factorization for a mixture becomes:

$$p_{XY}(\boldsymbol{x},\boldsymbol{y}) = \sum_{j=1}^{K} \Big[\pi_j \mathcal{N}_{Y|X,j}(\boldsymbol{y}; m_j(\boldsymbol{x}), R_{Y|X,j}) \quad (5)$$

$$\times \mathcal{N}_{X,j}(\boldsymbol{x};\boldsymbol{\mu}_{X,j}, R_{XX,j})\Big], \quad (6)$$

with

$$m_j(\boldsymbol{x}) = \boldsymbol{\mu}_{Y,j} + R_{YX,j}R_{XX,j}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}). \quad (7)$$

$m_j(\boldsymbol{x})$ defines one of the above mentioned MoE $\mathbb{R}^p \mapsto \mathbb{R}^q$ expert functions which in our GMM case are q linear functions. The slope is defined by $R_{YX,j}R_{XX,j}^{-1}$. If steered, i.e. non-homogeneous Gaussians in GMM are used, our desired linear steering kernels for SMoE are obtained. Each kernel adapts to local statistics but - in contrast to RBFs, SVR and SKR - each kernel has global support over the entire signal domain.

The MoE approximation function is derived from the conditional pdf $Y|X$ [38]

$$p_Y(Y|X=\boldsymbol{x}) = \sum_{j=1}^{K} w_j(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}; m_j(\boldsymbol{x}), R_{Y|X,j}), \quad (8)$$

with mixing weights

$$w_j(\boldsymbol{x}) = \frac{\pi_j \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{X,j}, R_{XX,j})}{\sum_{i=1}^K \pi_i \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu}_{X,i}, R_{XX,i})}. \quad (9)$$

Note that the MoE gating function in Eq. (9) corresponds to the normalized exponential or the *softmax* function frequently used in artificial neural networks. It defines the support region for each kernel and ensures that each sample has support.

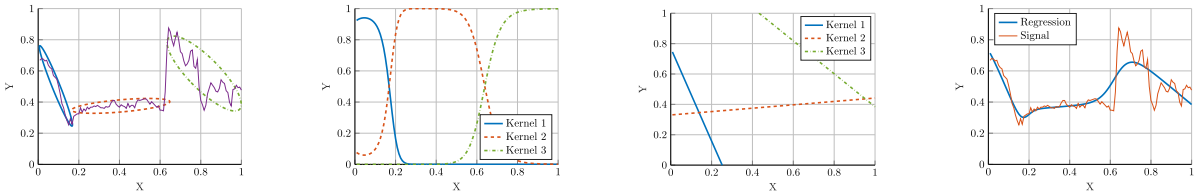
We enable non-linear regression by adopting the expected value $\hat{\boldsymbol{y}}$ given a sample location $\boldsymbol{x}$ through the conditional. From Eq. (8) and (9) follows the *non-linear regression function* $m(\boldsymbol{x})$:

$$\hat{\boldsymbol{y}} = m(\boldsymbol{x}) = \mathbf{E}[Y|X=\boldsymbol{x}] = \sum_{j=1}^{K} w_j(\boldsymbol{x})m_j(\boldsymbol{x}). \quad (10)$$

The trustworthiness of the prediction of the ith component in Y, can then be evaluated by calculating the prediction variance $\mathrm{var}[Y^i|X=\boldsymbol{x}]$.

C. Robust Modeling of GMMs for SMoE

The previous section showed how to perform smoothed piecewise regression based on a GMM. The GMM models the joint probability density function of both X and Y, which in our approach correspond to the pixel coordinates and the pixel amplitudes. The question remaining is how to estimate the GMM parameters given image pixel data.



(a) Signal and 2-D Gaussians (b) Weights/Gating (c) Linear regressors (d) Regression vs. signal

Fig. 2. A 1-D regression example using SMoE on a part of a scanline taken from *Lena* using three Gaussian kernels ($K = 3$). The 2-D GMM models the joint probability between X and Y (a). From this model, a gating function (b) and the regressors (c) are derived. The regressors are summed according to the gating function to yield the regressed function in (d).

The *Expectation-Maximization* (EM) algorithm is frequently used for estimating parameters Θ of a mixture model in an unsupervised learning approach [39]. The EM algorithm iteratively maximizes the loglikelihood, which in the case of the joint probability of X and Y given a mixture model of exponential distributions is given by

$$l(\Theta|X,Y) = \mathbf{E}[\log p(\boldsymbol{x},\boldsymbol{y}|\Theta)] \tag{11}$$

$$= \frac{1}{N}\sum_{i=1}^{N}\log\sum_{j=1}^{K}\pi_j p_{XY}(\boldsymbol{x}_i,\boldsymbol{y}_i|\theta_j). \tag{12}$$

In each iteration k, first the soft-membership $\hat{z}_{ij}$ of a pixel $i \leq N$ to each Gaussian $j \leq K$ is estimated, i.e. the likelihood of that sample originating from that Gaussian (Expectation step, or E-step). Secondly, based on these soft-memberships the kernel parameters are re-estimated based on the pixel data that belong to that cluster (Maximization step, or M-step) [40]:

$$\text{(M-step)}\ \Theta^{k+1} = \arg\max_{\Theta}\hat{z}_{ij}. \tag{13}$$

As such, the Gaussians are iteratively fit onto the data. The optimization problem is unfortunately non-convex and converges to a local optimum [41]. Consequently, EM is sensitive to the initialization of the parameters of the Gaussians, i.e. the positions and steering. Furthermore, it is clear that this formulation scales badly in terms of complexity. We can divide the problem into smaller blocks as in Section V-B, but even then for light field data, these blocks possibly consist of millions of pixels. Previous works on SMoE always relied on the batch version as described above. However, in the application to light fields, our old approach suffered from robustness issues when the amount of kernels became large [7]. The more kernels that are added, the more the optimization becomes sensitive towards local optima due to the vastly increasing number of parameters to optimize.

In order to mitigate these issues, a stochastic online version of the EM algorithm, or *minibatch* EM was proposed [42], [43]. In minibatch EM, parameters are updated in a stochastic fashion by taking random minibatches of size m (randomly sampled pixels) and performing the M-step according to a learning speed η:

$$\Theta^{k+1} = (1-\eta)\Theta^{k} + \eta\left(\arg\max_{\Theta}\hat{z}_{ij}\right) \tag{14}$$

By using minibatches, the local optimum change in every iteration. As such, it converges more easily to a solution closer to the global optimum. Furthermore, as $m \ll N$, each iteration takes up N/m times less memory in the E-step and substantially lowers the duration of a single iteration. Section VI-B provides experimental evidence for these claims.

D. Example: 1-D Steered Mixture-of-Experts (SMoE)

For illustration purposes, Fig. 2 depicts a SMoE regression of samples from a 1-D image scan line. The Gaussians/kernels were optimized using the EM algorithm. Notice that both X and Y are 1-D, we thus estimate 2-D pdfs using steered Gaussians. In Fig. 2a, the Gaussians in the GMM are visualized as ellipses, which indicate iso-probability. Each Gaussian is responsible for a region in X defined by the weights (Eq. 9), as visualized in Fig. 2b. Fig. 2c shows that each kernel also yields a linear regressor based on the expectation of the conditional $Y|X = \boldsymbol{x}$. Finally, the resulting *smoothed piecewise linear* regression function by the weighted sum over all kernels is shown in Fig. 2d.

IV. SMoE for 2-D Image, 4-D LF Images, and 5-D LF Video Representation

A. Introduction

In this section, we outline the application of the SMoE regression approach on 2-D images, and provide a number of illustrative results. Consequently, this provides easier understanding of the SMoE framework before introducing the extension to 4-D LF images and 5-D LF video in Section IV-C.

For grayscale images, we define $\boldsymbol{x}_i \in \mathbb{R}^2$ as the locations and $\boldsymbol{y}_i \in \mathbb{R}^1$ as the amplitudes of image pixels. Regressing the model is equal to finding the expected amplitude $\hat{\boldsymbol{y}}_i$ given a location $\boldsymbol{x}_i = [x_{i,1}, x_{i,2}]$ through the "learned" conditional pdf, i.e. $\hat{\boldsymbol{y}} = m(\boldsymbol{x})$. Each kernel defines a linear expert function $\mathbb{R}^2 \mapsto \mathbb{R} : m_j$ as their regressor, which visually describes a gradient per kernel. The gradient indicates how the signal behaves around the center of the kernel (Eq. 7). Furthermore, each kernel defines a 2-D window gating function $\mathbb{R}^2 \mapsto \mathbb{R} : w_j$, which defines the operating region, or support of the expert. The window function w_j gives weight to each sample, indicating the soft membership of that pixel to that component (Eq. 9). Note that by jointly modeling the pixel locations and amplitudes, our kernel windows can steer along edges and adapt to regional signal intensity flow, similar to the locally-supported SKR [27].

When extended to support and regress color values in images, the output Y becomes a 3-D random variable (e.g. in case of the YCbCr color space). In this case the steered kernels are based

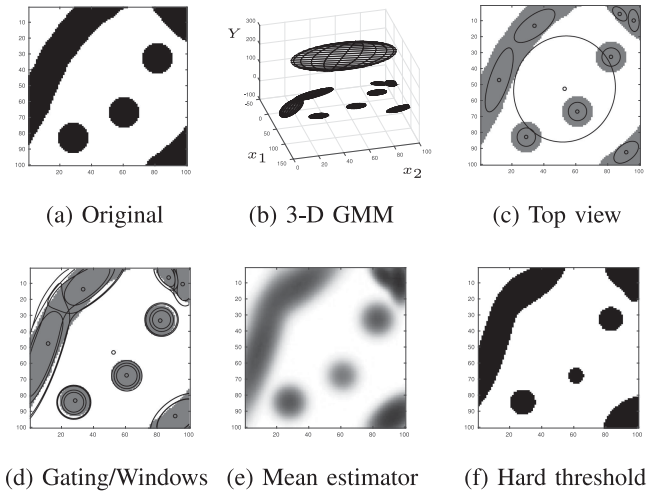


Fig. 3. Example of a black-white image (a) modeled by 9 kernels for 10.000 pixels (1 kernel covers ± 1111 pixels on average). The kernels are visualized in (b) in $\mathbb{R}^3$ joint coordinate and color space. In (c) the spatial spread of the kernels is shown as an overlay on the original image. Illustration (d) shows the mixing weight w_j (or responsibility) of each kernel j on each pixel after softmax. The continuous regression is shown in (e) and the regression quantized into 1 bit in (f). Note how the kernels in (b) are virtually flat in the Y dimension as they represent constant colors.

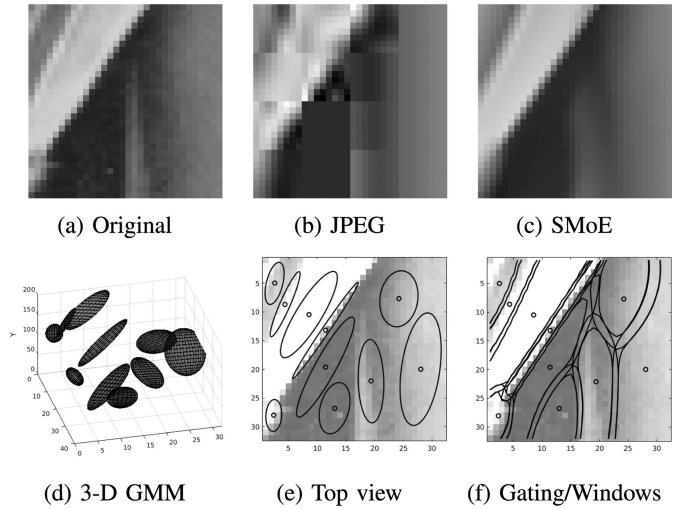


Fig. 4. Example of SMoE modeling and reconstruction on a 32×32 pixel crop from *Lena* (a). The kernels are visualized in joint coordinate and color space $\mathbb{R}^3$ (d) and as an overlay to the image only showing the spatial spread in X (e). (f) illustrates how these kernels are responsible for irregularly-shaped regions after softmax. At the same bitrate, JPEG (b) results in block artifacts whereas SMoE has a reconstruction (c) that is smoothed along image features. Note how each component covers a range of luma values in Y and the corresponding regressors thus result in gradients.

on a "learned" 5-dimensional pdf (2-D location, 3 color channels), although the regression for each channel is independent to each other. The 5-D kernels now explore correlation in horizontal and vertical dimensions as well as in 3-D color space. In consequence, each color channel has the same 2-D window w_j (Eq. 9), but different and independent regressors $m_{Y,j}$ (luma), $m_{Cb,j}$, and $m_{Cr,j}$ (chroma):

$$m_{Y,j}(\boldsymbol{x}) = \boldsymbol{\mu}_{Y,j} + R_{YX,j} R_{XX,j}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}), \tag{15}$$

$$m_{Cb,j}(\boldsymbol{x}) = \boldsymbol{\mu}_{Cb,j} + R_{CbX,j} R_{XX,j}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}), \tag{16}$$

$$m_{Cr,j}(\boldsymbol{x}) = \boldsymbol{\mu}_{Cr,j} + R_{CrX,j} R_{XX,j}^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_{X,j}). \tag{17}$$

In the following we briefly illustrate the functioning and properties of SMoE on binary, gray-scale and color images, and, finally, 4-D light fields.

B. 2-D Image Examples

We use the binary image in Fig. 3a to illustrate the approximation of the binary pixel values using only a very small number of kernels ($K = 9$). The GMM (after learning using EM) results in the 3-D mixture model illustrated in Fig. 3b and 3c. Due to the fact that only two luma values are present in the image, the estimated 3-D ellipsoids are flat along the Y-dimension, i.e. $R_{YY,j}$ and $R_{YX,j}$ are zero for each component j. From Eq. (7) and Eq. (4) results that the regressors defined by each kernel are constant planes, and the conditional variance is zero. Also note that the full background is covered by a single large white kernel. While all other experts are gated to provide only local support within this image, one expert provides local as well as global support.

The gating windows are shown in Fig. 3d and confirm the directional steering operation of the experts. The windows softly overlap while forming arbitrarily-shaped segments. When the expected value of the conditional of the mixture model is calculated (*mean estimator*), we arrive at a continuous-tone image shown in Fig. 3e. A binary image can be yielded in two ways: (1) by hard thresholding as illustrated in Fig. 3f, i.e. mapping all luma values $y \geq 0.5$ to be white, and all luma < 0.5 to be black, or (2) by using the mode of the conditional pdf (*mode estimator*). Even though we only used nine kernels to represent the image content, it is clear that all "objects" are represented. It is apparent that image approximation using SMoE results in geometrical distortion of image objects.

Fig. 4 illustrates the modeling and reconstruction of a 32×32 pixel crop of *Lena* using $K = 10$ components. The SMoE model parameters were quantized prior to reconstruction to arrive at a designated bit rate in order to allow a comparison with JPEG (Fig. 4b) as a simple compressed and coded representation. For fair comparison, the bits required for the JPEG header were subtracted. Both representations are at around 0.35 bit/sample [5]. Comparing Fig. 4b and 4c, it is apparent that especially the edges are reconstructed with impressive quality and sharpness by our approach. It can be argued that SMoE provides for a much more efficient and sparse image representation than JPEG for this type of image content. Fig. 4d shows the steering of the 3-D ellipsoid Gaussian "cigar" components, which define the m_j "global" 2-D steering planes (gradients) for regression. Fig. 4e illustrates steering of the ellipsoids projected onto the 2-D pixel domain. It is apparent the SMoE kernels harvest directional pixel correlation.

The respective window functions dictate how the kernel gradients are gated. The windows overlap adaptively into adjacent image regions and enable either smooth transitions between regions, or abrupt changes. The windows are of arbitrary shape and steer along edges. This assures that dominant edges are well reconstructed considering the low amount of components. Note that the dominant gradient on the right is very well approximated by a single kernel. Fine details and noise are eliminated which

Fig. 5. An example of mean estimated reconstructions of a 128 × 128 image. Original (left) followed by models with 25, 100, 250, 750, and 2000 components, i.e. ranging from 1 kernel covering ±655 to ±8 pixels on average. It is clear that SMoE provides a continuously-refined low-pass version of the original.

is the result of the very sparse representation with only 10 components. Fig. 5 shows the extension towards color images with SMoE models with varying levels of sparsity (number of components K between 25 and 2000). Fig. 5 thus illustrates that SMoE provides a continuously-refined low-pass version of the original.

One of the most interesting properties is that SMoE yields a representation that is a continuous parametric closed-form expression. Furthermore, the kernel centers are not limited by a sampling grid and the kernel steering can have subpixel precision. SMoE can therefore sample anywhere on the image plane and thus has intrinsic interpolation, resampling, and super-resolution with sharp edges readily available. Another feature is that the kernel parameters provide novel image descriptors, e.g. the steering parameters, local gradients, intensity flow [5]. Furthermore, the gating functions provide a segmentation of the image. These tools are readily available for several (decoder) post-processing tasks. This may include tasks such as segmentation, noise reduction, scale conversion, image similarity retrieval, classification and object recognition, to name a few. An elaborate discussion on the extended additional functionality is beyond the scope of this article.

C. 4-D Light Field Representation

The 4-D light fields considered in this section are short-baseline light fields resulting from lenslet-type cameras. However, the theory does not rely on any hardware assumptions and is thus applicable to LFs from any acquisition source. In the following, we adopt the LF parametrization $\mathrm{LF}(a_1, a_2, x_1, x_2) = (\mathrm{Y}, \mathrm{Cb}, \mathrm{Cr})$, in which (a_1, a_2) are the camera (row, column)-coordinates on the camera plane and (x_1, x_2) are the pixel (row, column)-coordinates on the image sensor. This is conform with the data structure that is yielded by the LF Toolbox v0.4 [44]. Consequently, our GMM is 7-D, with the X-coordinate being 4-D and the Y-amplitude being 3-D. Consequently, the soft-windows $w_j(\boldsymbol{x})$ (Eq. 9), describe a 4-D volume per kernel, and the expert function $m_j(\boldsymbol{x})$ (Eq. 7) describes for each color channel a 4-D gradient, i.e. a linear function from $\mathbb{R}^4$ to $\mathbb{R}$.

Fig. 6a shows a small LF, including the *epipolar-plane images* (EPI) on the bottom and right side. The red lines indicate where the 4-D space is sliced, i.e. indicating where the EPIs are located spatially. As the kernels are likelihood optimized, they are expected to steer along the diagonal EPI structures. As such, kernels can be responsible for different pixel coordinates (x_1, x_2) depending on the camera coordinate (a_1, a_2). Visually, it seems thus that kernel windows move over the image plane when moving the viewpoint. The magnitudes of these shifts correspond to the slopes within the EPIs. Interestingly, these

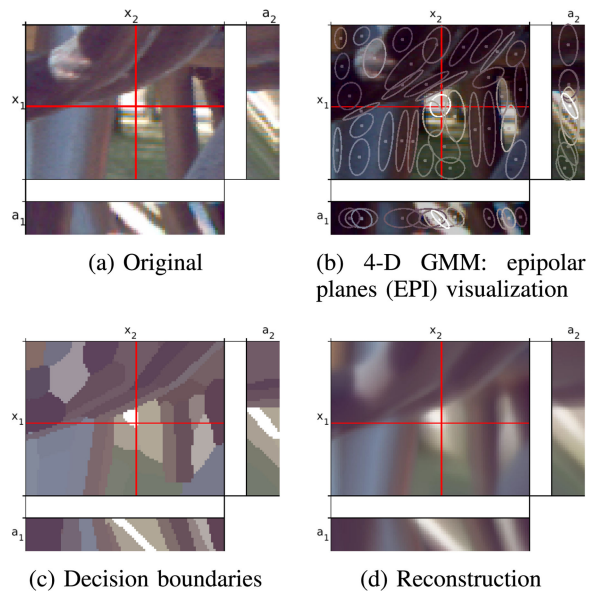


Fig. 6. SMoE modeling and reconstruction of a cropped LF (*I01 Bikes* [45] [46]) using a very low amount of kernels for visualization purposes (K = 35). The original is shown in (a). The kernels are visualized in the 4-D coordinate space with camera coordinate dimensions (a_1, a_2) and pixel coordinate dimensions (x_1, x_2) in (b). The EPI images are shown below and right of the spatial crop, corresponding to the pixels indicated by the red lines. The reconstruction is shown in (d). Note that SMoE implicitly provides a 4-D consistent segmentation (c), when we indicate for each pixel $\boldsymbol{x}$ which kernel j is dominant (highest $w_j(\boldsymbol{x})$). The segmentation illustrates how the kernels are steered along the EPI diagonal lines.

slopes are proportional to the depth of that point in the scene [31]. The orientation of the kernels along the EPIs thus implicitly codes depth and could potentially be used as a depth estimator [7]. Furthermore, a single kernel can yield different color values when viewed from a different camera coordinate through the 4-D gradient. As such, it allows us to model non-Lambertian reflectance.

Fig. 6b shows a low order GMM fit onto the data, note that our kernels have a spread in all four X dimensions simultaneously. Fig. 6c illustrates the segmentation, which is nothing more than the hard-decision of our soft-windows $w_j(\boldsymbol{x})$. It is clear that our windows steer along the EPI structure and soft-partition the entire 4-D space, thus yielding global support. Using Eq. ((10)), the reconstruction is illustrated in Fig. 6d. Fig. 7b shows the reconstructed (7,7)-view from the *I01 Bikes* LF shown in Fig. 7a [45]. The modeling is detailed in Section V. Note how the rust speckles turn into smudges in the reconstruction, which could arguably be seen as a visually-pleasing quality decay. This is however heavily penalized when using objective metrics such as *Peak-Signal-to-Noise Ratio* (PSNR) and *Structural Similarity*

(a) Original view (b) Reconstruction

Fig. 7. *I01 Bikes* [45] [46] light field example (K = 8960), showing a central view with $(a_1, a_2) = (7, 7)$ with mean $\text{PSNR}_{\text{YCbCr}}$: 30.71 dB and mean SSIM_{Y}: 0.86 (objective evaluation as in [45]). Note the smoothing artifacts in (b) which originates from kernels being responsible for a large number of pixels. For example, the mud speckles on the "peugeot" bar in (a) turn into smudges in (b).

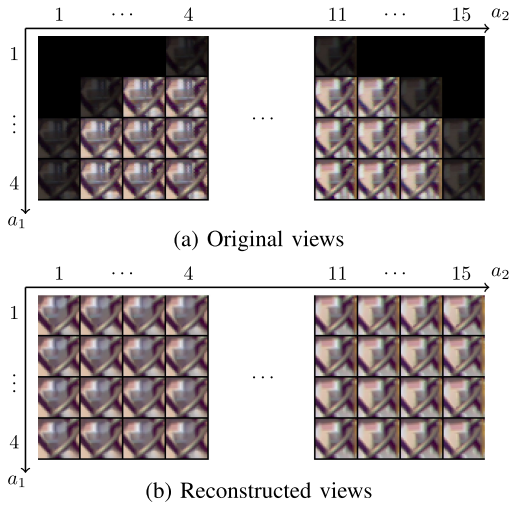


Fig. 8. In the original light field there are missing views due to the sensor architecture. Views that are at the outer edges of the camera plane are shown in black, with a_1 and a_2 corresponding to row and column in the camera plane. However, view reconstruction using SMoE (26 × 27 spatial crop from *I02*) shows consistent extrapolation of these outer views.

(SSIM) [47]. Note that, the reconstruction is slightly 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 SMoE is able to reconstruct views that were not captured, which is in stark contrast to dense representations that require a separate view synthesis process. Our model has a continuous representation, as such any view in the domain can be readily reconstructed. Limited extrapolation is also possible. Fig. 8a, shows that the LF data structure (obtained through the LF Toolbox [44]) results in black views in the corners of the camera plane. 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. 8b). However, extensive evaluation is considered out of scope.

D. 5-D Light Field Video Representation

Light field videos are LFs captured at intervals over time and thus yield a 5-D coordinate space (t, a_1, a_2, x_1, x_2). The theory remains identical and the (a_1, a_2, x_1, x_2) keep the same properties identically to 4-D LF image models, only a time dimension t is added. However, the time dimension does behave very differently compared to the camera-plane dimensions and the spatial dimensions. In practice, the kernels are all maximally elongated along the EPI strips while having a rather limited spatial spread. Interestingly, along the time dimension both are commonly present. Kernels that represent the light irradiated by a static or a linearly-moving object will have a large spread along the time dimension. However, in the case of non-linear movement or rapidly changing color values, kernels will be short along the time dimension. This needs to be taken into account during modeling as an adequate kernel density over the whole 5-D coordinate space is desired. In most cases, the frames for each viewpoint are synchronized during acquisition or are resampled before processing. However, as our representation is resolution agnostic, frames can be captured at irregular intervals. Only the absolute timestamp t of the frame is necessary for our modeling process. Furthermore, as our model is continuous, we can reconstruct all views at synchronous timestamps without any other methods involved.

The main challenge in LF video is the incredible number of samples that need to be modeled. A LF video with 10 × 10 viewpoints, in full HD at 30 fps thus yields 6,220,800,000 pixels per second! Sparse representations such as SMoE are hugely beneficial for such higher-dimensional modalities as a single kernel can span over a large number of pixels spread out over five dimensions simultaneously. Table I shows the number of kernels K for the models used in the coding experiments in Section VI. We observe the maximum K to be 33,415 for the *train1* LF video sequence which consists of 1,257, 242, 624, i.e. over 1 billion original samples. One SMoE kernel thus covers 37,625 original pixel on average while resulting in a reconstruction with 0.96 SSIM and 36.24 dB PSNR. To illustrate, the previously shown examples showed an average pixel-coverage per kernel to be 8 to 655 pixels for the presented 2-D image (Fig. 5) and an average pixel-coverage of $\pm 4,630$ pixels for the 4-D LF image example (Fig. 7). This is one of the strengths of such a sparse representation, i.e. whereas the number of pixels grows exponentially with p, the number of required kernels has more of a linear relation and thus performs better as p increases. The exact relation is not defined as the number of required kernels depends on the image content.

Analogously, the descriptors that were enabled by SMoE for 4-D LF images (e.g. depth and 4-D segmentation) are similarly available for 5-D LF video, e.g. 5-D segmentation and rough depth at each point in time. Furthermore, the orientation of the kernels are likely to follow motion as the modeling process steers the kernels to harvest correlation over time. Extensive evaluation is considered future work.

V. Proposed Coding Scheme

A. Introduction

Our coding strategy is illustrated in Fig. 9. First, kernels are fit onto the LF data during modeling. The modeling thus results in a set of parameters per kernel j, as described in Section IV-C. However, not all kernel parameters are needed for reconstruction, or can be fixed. Such parameters are thus not coded in

TABLE I
CODING PARAMETERS, OBJECTIVE AND SUBJECTIVE RESULTS

	bpp / kbps	K	L	b	σ_T	σ_A	σ_Y	σ_{UV}	σ_s	σ_{XY}	σ_{TY}	σ_{AY}	PSNR (dB)	SSIM	MOS	CI
I01	0.006 bpp	1393	1024	12	-	0.50	1.00	0.12	0.12	0.50	-	1.00	27.37	0.756	1.90	0.149
	0.030 bpp	17188	1024	10	-	0.50	0.50	0.12	0.50	0.50	-	0.25	31.24	0.873	3.77	0.181
	0.098 bpp	30755	4096	14	-	0.50	0.25	0.25	0.50	0.12	-	0.50	32.68	0.898	4.00	0.164
	0.276 bpp	118846	4096	12	-	0.50	0.25	0.12	0.12	0.50	-	1.00	33.21	0.906	4.16	0.143
I02	0.006 bpp	1386	1024	12	-	0.50	0.50	0.25	1.00	1.00	-	0.50	25.55	0.629	1.52	0.138
	0.028 bpp	17202	64	10	-	0.25	0.50	0.25	0.25	0.12	-	0.50	28.90	0.795	2.97	0.223
	0.092 bpp	31946	2048	14	-	0.25	1.00	0.12	0.12	0.25	-	0.12	31.16	0.864	4.03	0.105
	0.346 bpp	121649	4096	14	-	0.25	0.50	0.12	0.25	0.12	-	0.50	32.16	0.884	4.29	0.145
I03	0.006 bpp	3455	64	10	-	0.25	0.25	0.12	0.25	0.12	-	0.50	26.00	0.643	1.90	0.196
	0.030 bpp	17208	64	10	-	0.25	1.00	0.12	0.25	1.00	-	0.25	28.48	0.790	2.84	0.190
	0.076 bpp	32146	4096	12	-	0.50	0.25	0.12	0.50	0.25	-	0.12	30.66	0.863	3.94	0.233
	0.280 bpp	121720	2048	12	-	0.25	1.00	0.25	1.00	1.00	-	0.25	31.56	0.885	4.13	0.205
cats	247.48 kbps	7915	4096	12	0.12	0.12	0.50	0.25	0.25	0.25	0.25	0.25	33.37	0.943	-	-
	479.16 kbps	11454	4096	12	0.50	0.12	0.25	0.12	0.12	0.50	1.00	0.25	34.71	0.952	-	-
	845.48 kbps	23655	4096	12	0.50	0.12	0.50	0.12	0.25	1.00	0.25	0.50	35.91	0.960	-	-
	1056.80 kbps	33415	4096	12	0.12	0.12	0.25	0.12	0.25	0.50	1.00	0.25	36.24	0.962	-	-
train1	244.91 kbps	6475	1024	10	0.50	0.12	0.50	0.12	0.50	1.00	0.50	0.50	33.14	0.935	-	-
	484.98 kbps	9379	4096	10	0.50	0.12	0.25	0.25	0.50	0.50	0.50	1.00	34.13	0.946	-	-
	807.09 kbps	19230	4096	10	0.50	0.50	0.25	0.25	0.50	1.00	0.25	1.00	34.09	0.952	-	-
	1240.11 kbps	26954	4096	12	0.50	0.12	0.50	0.25	0.50	0.50	0.25	0.50	35.02	0.957	-	-

Coding parameters, objective (PSNR, SSIM) and subjective (MOS, Confidence Interval CI) quality results for both 4-D LF images and 5-D LF video SMoE models. The models contain K kernels. The covariance matrix R_{XX} is normalized by dividing the scale $s = |R_{XX}|^{1/p}$. The normalized R_{XX} is then coded using a dictionary with L entries. The identifier of the closest kernel in that dictionary is then coded using $\log_2(L)$ bits. The dictionary is trained on the set of normalized R_{XX} of this specific model and is transmitted along with the other parameters. All other parameters are normalized to have zero mean and a certain standard deviation σ depending on the parameter type. We then quantize using a fixed quantization step by dividing the maximum range into 2^b steps. All σ are ≤ 1. The rationale is that parameters with $\sigma < 1$ are quantized with more distortion, in order to save bits on these less important parameters. σ is 1, Except for $\sigma_T, \sigma_A, \sigma_{UV}$, corresponding to the components of the prediction error e corresponding to time (e_T), camera coordinates (e_A), luma center (e_{Y_Y}), and chrome centers ($e_{Y_{Cb}}, e_{Y_{Cr}}$). Next, $\sigma_s, \sigma_{XY}, \sigma_{TY}, \sigma_{AY}$ are the σ-values for respectively the scale s per kernel, the covariance between the luma channel and spatial, time, and camera coordinates. All parameters are then arithmetic coded, assuming a Laplace distribution as initialization.

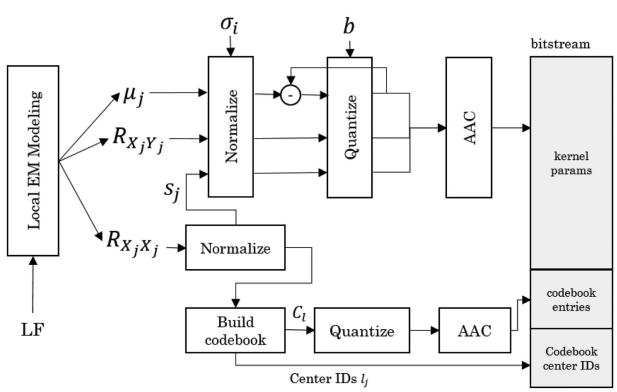


Fig. 9. The block diagram of the proposed encoding process. The 4-D LF is modeled in a blockwise manner and results in the parameters $\theta_j = \{\boldsymbol{\mu}_j, R_{XX,j}, R_{XY,j}\}_{j=1}^K$ for all K kernels (other parameters are not coded). Firstly, all except $R_{XX,j}$ are coded similarly, each parameter type i is normalized to have zero mean and a standard deviation σ_i (according to the importance of that parameter). These parameters are then quantized and coded in a single bitstream using an adaptive arithmetic coder (AAC). Secondly, $R_{XX,j}$ is first scaled by $1/s_j$, with $s_j = |R_{XX,j}|^{1/4}$ so that each determinant equals one. The normalized $R_{XX,j}$ are then used to build a codebook with centers C_l. The codebook centers C_l are then quantized and arithmetic encoded. Finally, for each kernel j, we encode the index l_j of the nearest cluster C_l.

order to save bits. We discard $R_{YY,j}$ as it is not necessary for the reconstruction (however, this removes the ability to calculate prediction variance at the decoder side). Furthermore, the priors π_j are assumed uniform, and are thus fixed to $1/K$ for 4-D LF images. Whereas, the π_j were coded similar to the gradient covariance coefficients for 5-D LF video. Also, the human visual system is less sensitive to changes in color than to changes in luminance. Therefore, the chroma slope components $R_{XY_{\text{Cb,Cr}},j}$ are assumed zero.

Secondly, the covariance matrices of the kernels in the coordinate space (i.e. $R_{XX,j}$) are highly redundant and a codebook approach is proposed. Finally, all other parameters are normalized, quantized and entropy coded. However, the centers $\boldsymbol{\mu}_j$ are quantized in a difference-coded method along an approximation of the shortest path. Note that both the modeling and the coefficient quantization contribute to the approximation error.

B. Local Modeling

The EM algorithm is used to estimate the parameters $\theta_j = (\pi_j, \boldsymbol{\mu}_j, R_j)$ for each component j [48]. In order to lower the computation demands, modeling is performed using a divide-and-conquer strategy. For 4-D LF images we chose to divide the LF into 4-D overlapping blocks that are independently modeled. Overlapping blocks mitigate block-artifacts that were present in [5]. Furthermore, in contrast to [5], all blocks in this work received the same budget of kernels. As such, we limit the number of free parameters. The blocks range over the full camera plane (a_1, a_2), but are limited on the image sensor plane, i.e. over the (x_1, x_2) dimensions. The reason here is that we expect kernels to have a large spread along the camera dimensions (aligning with the EPI lines), and limited spatial spread. A similar modeling strategy was used for 5-D LF video using minibatch updates.

C. Window $R_{XX,j}$ Quantization

As shown in Fig. 10, there is a high level of redundancy in the shapes of the kernels which are defined by the covariance in the coordinate space, i.e. $R_{XX,j}$. Therefore, we employ a

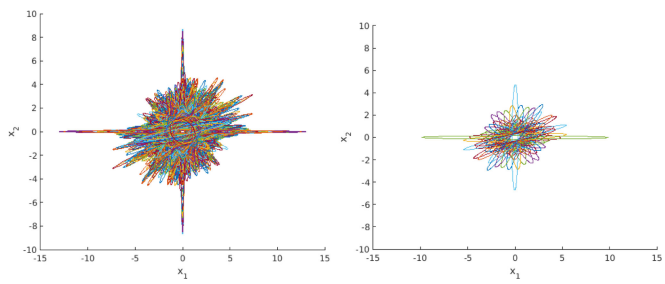


Fig. 10. On the left side, we illustrate all normalized $R_{XX,j}$ of a single 4-D LF image model ($K = 8960$) as ellipses. The $R_{XX,j}$ is the covariance in coordinate dimensions and defines the spread of each kernel in the coordinate space. For illustration purposes, we show only the two spatial dimensions (x_1, x_2). It is clear that there is a high level of redundancy. In order to reduce the redundancy in possible kernel shapes, we developed a codebook algorithm. The right plot illustrates the resulting codebook with only 64 dictionary entries. Codebooks are trained and binarized per model.

vector quantization-like method for coding the window covariance $R_{XX,j}$. 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. Thus, instead of coding the 4x4 matrix $R_{XX,j}$, we need to encode three items: (1) the smaller codebook with L entries $C_l, l \leq L$, (2) we code the index l of the closest cluster center for each kernel j and (3) a scale s_j for each kernel, as the codebook entries are normalized. The assumption is that similar reconstruction quality can be achieved with $L \ll K$.

We normalize all $R_{XX,j}$ by $|R_{XX,j}|^{1/p}$. As such, the constructed codebook contains normalized shapes with a determinant of one. The coding of the magnitude of the shape, i.e. $s_j = |R_{XX,j}|^{1/p}$ is discussed in the next subsection.

The KL-divergence for multivariate Gaussians $A \sim \mathcal{N}(\boldsymbol{\mu}_A, R_A)$ and $B \sim \mathcal{N}(\boldsymbol{\mu}_B, R_B)$ is given by

$$\mathrm{D_{KL}}(A \,||\, B) = \frac{1}{2}\left[\log\left(\frac{|R_A|}{|R_B|}\right) - d + \mathrm{tr}(R_B^{-1}R_A)\right] + \frac{1}{2}\left[(\boldsymbol{\mu}_B - \boldsymbol{\mu}_A)^T R_B^{-1}(\boldsymbol{\mu}_B - \boldsymbol{\mu}_A)\right] \quad (18)$$

As our data is normalized, $|R_A|$ and $|R_B|$ equal one. Furthermore, the windows are assumed to be centered on the origin, i.e. $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$ are zero. In order to obtain a symmetric similarity measure, we define our distance as

$$\mathrm{d}(A, B) = \frac{\mathrm{D_{KL}}(A \,||\, B) + \mathrm{D_{KL}}(A \,||\, B)}{2} = \frac{1}{4}\left(-2p + \mathrm{tr}(R_B^{-1}R_A) + \mathrm{tr}(R_A^{-1}R_B)\right)$$

Covariances are clustered around a centroid using $\mathrm{d}(A, B)$. At each iteration, the new centroid covariance C_l is calculated as the mean covariance of the members of the cluster l, and renormalized.

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: $C_l = U^T U$. U is vectorized and each coefficient is coded analogously to the slopes $R_{XY,j}$ (see Section V-D). At decoder side, the multiplication $U^T U$ ensures the reconstructed covariance to be semi-positive definite again. At decoder side, $R_{XX,j}$ is thus reconstructed as follows:

$$\tilde{R}_{XX,j} = s_j \times C_{l_j}, \quad (19)$$

with l_j the index of the closest codebook center C to the original normalized kernel covariance matrix and is coded using $\log_2(L)$ bits per kernel.

However, the outlined algorithm is known to scale badly to high numbers of kernels. At each iteration, the distance is between each kernel's $R_{XX,j}$ and codebook center's C_l is calculated (cfr. Section III-C). Similar to the minibatch approach for the EM algorithm, we developed a minibatch codebook training method by employing a per-cluster learning rate as in [49]. This allows us to converge orders of magnitudes faster while requiring vastly less memory.

D. Center $\boldsymbol{\mu}$ and Slope $R_{XY,j}$ Quantization and Arithmetic Coding

The kernels are sorted by the centers $\boldsymbol{\mu} = [\boldsymbol{\mu}_X, \boldsymbol{\mu}_Y]$ by defining a path that comprises every component once in a greedy fashion. Start with the component j closest to the origin. Find component k $(k \neq j)$ so that $|\boldsymbol{\mu}_j - \boldsymbol{\mu}_k|$ is minimal. As such, each $\boldsymbol{\mu}_{j-1}$ is a good predictor for $\boldsymbol{\mu}_j$. We then choose to only quantize and binarize the prediction error. Note that the prediction error $\boldsymbol{e}_j$ is calculated based on the dequantized $\tilde{\boldsymbol{\mu}}_{j-1}$ in order to prevent error propagation.

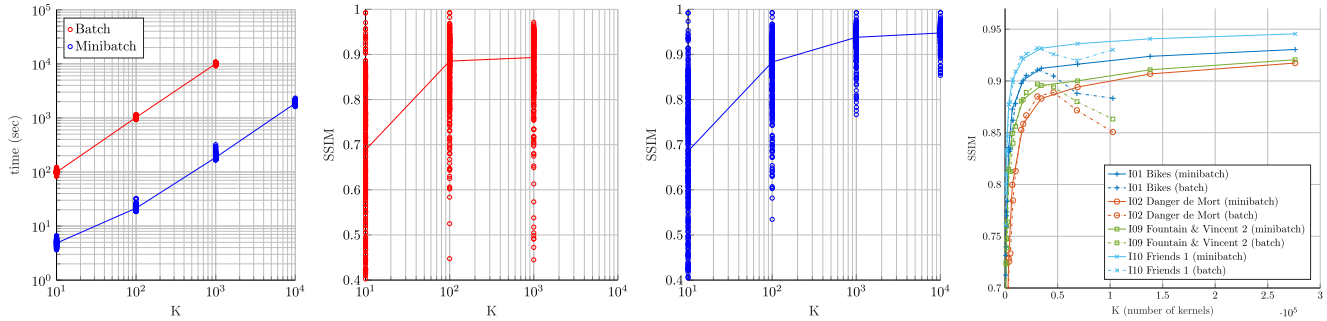
$$\boldsymbol{e}_j = \boldsymbol{\mu}_j - \tilde{\boldsymbol{\mu}}_{j-1} \quad (20)$$

This scheme is generally known as *Differential Pulse Code Modulation* (DPCM) and is illustrated by the feedback loop in Fig. 9. We thus arrive at a $(p + q)$-dimensional prediction error vector for each kernel j, i.e. 7-D and 8-D respectively for LF images and video. These vectors typically follow a Laplacian distribution.

Secondly, the full $p \times 3$ covariance matrix $R_{XY,j}$ is not entirely encoded. As we operate in the 3-D YCbCr space, we intend only to encode the gradients along the luma channel. We thus continue working with a $p \times 1$ covariance matrix, the other elements are assumed to be zero. From our tests, we observed that the remaining values naturally follow a Laplacian distribution. The final parameter to be encoded is the magnitude of the covariance matrix $R_{XX,j}$, which is $s_j = |R_{XX,j}|^{1/p}$. This parameter naturally follows a distribution close to a positive-Laplacian distribution.

For LF images and video, a total of respectively 12 or 15 parameters are thus encoded per kernel j, i.e. the prediction errors $\boldsymbol{e}_j$, the p dimensions in $R_{XY,j}$ (discarding the chroma dimensions) and the shape magnitude s_j. Furthermore, the priors π_j are also included for light field video. Due to these observations, we aim to encode all these parameters in a single *Adaptive Arithmetic Coder* (AAC) which assumes a Laplacian distribution. Therefore, we need to align all the distributions of the remaining 12 or 15 parameters. Furthermore, we want more distortion in less important parameters in order to save bits.

In order to align the distributions and to allow more distortion in some parameters than others, we propose the following.



(a) Time speedup 100x (b) SSIM batch (c) SSIM minibatch (d) Light field modeling

Fig. 11. Performance evaluation of batch EM vs. minibatch EM in terms of speed (a) and SSIM (b, c) on a dataset of light field crops. Note the logarithmic x-axis for (a), (b), and (c), as well as the logarithmic y-axis in (a). There are two things that are worth noting. First, an impressive x100 speed-up is visible in (a). Secondly, when comparing the samples for batch (b) and the samples for minibatch (c), it is visible that higher quality is more consistently achieved for minibatch. In contrast to the batch method (b), where the quality could be unacceptable for even a large number of kernels for certain samples. The reconstruction quality of complete light fields is validated in (11d). It is clear that the minibatch approach heavily increases the robustness for large Ks, while achieving a x100 speed-up.

We normalize the resulting parameters to have zero mean and a certain standard deviation σ_i, with $\sigma_i \leq 1$. We then quantize using a fixed quantization step $|min_val, max_val|/2^b$, based on the minimum and maximum value of all normalized parameters. The rationale is that parameters with $\sigma_i < 1$ are quantized with more distortion, in order to save bits on these less important parameters. Finally, we entropy code the quantized values by employing an adaptive arithmetic coder which is initialized by a Laplacian distribution.

To summarize, the following choices can be made in order to save bits. Note that all choices lead to a possible visual degradation of the reconstructed light field:

1) Modeling using a lower number of kernels K;
2) Decreasing the standard deviation σ_i of each parameter type i during normalization;
3) Increasing the quantization step size by decreasing b;
4) Decreasing the number of elements in the codebook.

VI. Approximation and Coding Experiments

A. Introduction

In this section, we firstly investigate the performance of our new robust modeling technique as introduced in Section III-C. We evaluate in terms of speed and reconstruction quality, compared to the modeling used in [7]. Secondly, we validate our proposed coding scheme using models trained using our robust modeling scheme using objective metrics. Thirdly, we evaluate the subjective quality of the coded light fields. Finally, the compression efficiency for light field video is investigated.

For these experiments, three datasets are used. First, a new dataset with 324 small light field crops was extracted from the EPFL lenslet dataset used for ICIP Grand Challenge and the Call for Proposals for JPEG Pleno [8]. The crops have 10-bit color depth, 64×64 image pixel resolution and 13×13 camera coordinates. Each block thus contains 692,224 samples. Second, five full LFs from the same EPFL dataset were used. Finally, two light field video sequences were used: *cats* and *train1* [50]. The LF videos resulted from temporally upscaling a lenslet camera to 30fps with two light field video sequences of resolution 512×352 for approximately 100 frames and 8×8 views.

B. Minibatch vs. Batch EM

As mentioned in Section III-C, the online EM introduces two new parameters: the batch size m and a driver for the learning rate α. From our experiments, we found that for large K, the reconstruction quality becomes sensitive towards the values of (m, α), with differences of up to 10 dB PSNR. A careful analysis of these parameters is thus advised. Empirically, we found the following parameters for blocks of $13 \times 13 \times 64 \times 64$: $m = 1000$, and $\alpha = 0.5$ when $K < 1000$, and $\alpha = 0.8$ when $K > 1000$. Both the batch and the minibatch approaches are implemented using MATLAB.

Fig. 11 shows results of the reconstruction quality and modeling speed of the above mentioned dataset. It is clear that for the same number of kernels K, the minibatch approach is up to 100x faster. Furthermore, the results confirm the increase of robustness. The desired behavior of having monotonous positive relation between number of kernel K and reconstruction quality is experimentally confirmed. Given 10,000 kernels (1 kernel per 92 pixels), the minibatch EM algorithm reaches up to 37 dB PSNR on average and 0.95 SSIM. Using the local block-based modeling (with 3-pixel overlapping blocks) in Section V-B, we compare the performance of the modeling on the full LFs. Firstly, we can clearly see that there is a strong increase of robustness. Whereas using the batch EM does not guarantee a monotonic increase of SSIM, the minibatch method does. Secondly, the strong decrease in runtime allows us to create models with a much higher number of kernels. We can conclude that, given careful a priori analysis of the hyperparameters, the usage of minibatches for training GMMs is beneficial in terms of speed, robustness and accuracy of reconstruction.

C. Light Field Image Coding

The following parameters were found using random search: blocksize, kernels per block K_i, quantization steps, and codebook size. The blocksize for the minibatch was fixed to 64 with K_i between 10 and 4000, whereas the blocksize for the batch EM ranged $[11, 17, 21, 32, 64, 128]$ with K_i between 6 and 48. The quantization step ranged [10,12,14], ratios $\sigma_i = [1, 1/2, 1/4, 1/8]$, and book sizes $L = [2^6, 2^8, 2^{10}, 2^{11}, 2^{12}, 2^{13}]$.

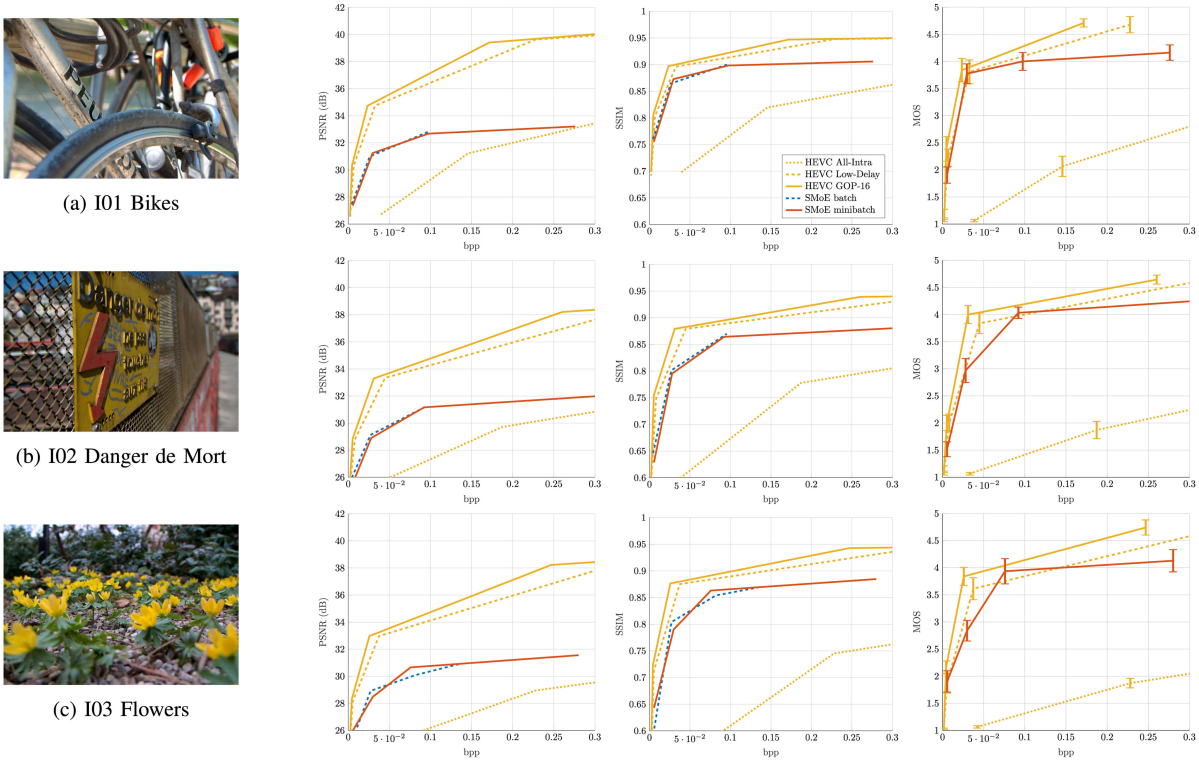


Fig. 12. Coding results comparing the three HEVC (All-Intra, Low-Delay, and GOP-16) and SMoE (batch and minibatch) in terms of PSNR, SSIM, and MOS scores. Notice the difference between the objective metrics (PSNR and SSIM) and subjective scores. It is clear that the distortions produced by SMoE (e.g. geometrical distortions, smoothing) are punished heavily by PSNR, and in lesser amount by SSIM. The SMoE distortions seem to have a visually pleasing effect as a MOS score of 4 indicates "Perceptual difference, but not annoying". Nonetheless, the loss of fine texture make it hard to achieve a MOS score closer to 5 "No perceptual difference". In general, we can say that up to and including a MOS score of 4, SMoE is competitive with motion-compensated pseudo-video coding of light fields using HEVC. Furthermore, note that PSNR and SSIM only capture the exact view reconstructions, whereas the MOS scores also captures smoothness between views and refocusing.

The largest portion of the computational complexity is situated in the local modeling and the codebook building and is very dependent on the number of kernels per block in modeling (see Fig. 11a). A model of 30 K kernels requires two hours, whereas 270 K kernels requires three days. The training of the codebooks depends on the number of kernels K and size of the codebook L, and range between some minutes and two hours. Note that this is non-optimized code in MATLAB running on a single thread of a IntelXeonCPU E5-2650 v3 @ 2.30 GHz machine. Reconstruction can be done in realtime [4].

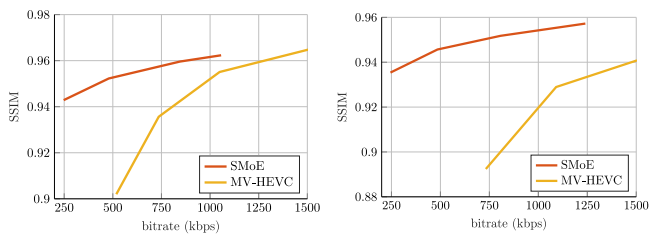
For comparison, we have encoded the LFs as HEVC videos using the reference encoder HM-16.17 [51]. In order to have a logical ordering, the video is built by traversing in a snakelike-manner from the top left view towards the bottom right view. In order to ensure a fair comparison, we did not encode the outer most views, i.e. $2 \leq a_1 \leq 14$ and $2 \leq a_2 \leq 14$, as they are not used in calculating the objective metrics. We compare three HEVC configurations ranging from granular random access (like SMoE) to low random access: HEVC All-Intra (GOP $= 1$), low-delay (GOP $= 4$), and with GOP $= 16$ with GOP being the *Group-of-Pictures*. For both GOP $= 4$ and GOP $= 16$, only a single I-frame is used in the first GOP, all following GOPs start with a P-frame.

Fig. 12 shows the rate-distortion (RD) curves for three LFs: *I01*, *I02*, *I03* optimized to SSIM. Table I shows the parameters and the metrics for each RD-point. We compare three HEVC configurations with batch- and minibatch-based SMoE. It is clear that for all images SMoE performs better than All-Intra HEVC with granular random access. However, SMoE is being consistently outperformed by motion-compensated HEVC. Batch and minibatch perform equally well, up until the point the batch-method does not allow higher kernel numbers.

Subjective tests were performed in order to assess a more general quality of experience, which aims to capture view reconstruction quality, view consistency, and refocus quality simultaneously. We exactly followed the recommended guidelines on passive subjective evaluation of light fields as in [17]. *Mean Opinion Scores* (MOS) were measured using a *Double Stimulus Impairment Scale* (DSIS), i.e. showing both the ground truth and the compressed sequence side-by-side. Four RD-points of the three HEVC configurations and for the minibatch SMoE method were selected in the lowest range, as this was assumed to cover the highest variance in MOS scores. Eleven refocused images were calculated using the *LFFiltShiftSum* function of the Matlab light field toolbox [44], the same slope values were used as suggested in [17].

The participant was not able to interact with the content, but a video was constructed for each RD-point that traverses the LF going through 97 selected viewpoints in a snake-like manner at 10 frames-per-second (fps) [17]. Next, the eleven refocused images were shown in an animation of 4 fps, going from a focused foreground to a focused background and back. The participant was asked to rate the compressed sequence on a scale: 1 (Very Annoying), 2 (Annoying), 3 (Slightly annoying), 4

Fig. 13. The setup used for the subjective experiments showing the 1080p Barco LC-47 monitor at eye-height. Both the ground truth and compressed sequences were shown side-by-side at native resolution.



(a) cats (b) train1

Fig. 14. Rate-distortion performance of MV-HEVC versus SMoE for two different light field video sequences. It is clear that SMoE provides high bitrate savings up to a factor of 4x.

(Perceptible but not annoying), and 5 (Imperceptible). The experiment was done in two sessions. Each session showed 40 stimuli side-by-side ($\pm$15 min per session) in a dark controlled environment as shown in Fig. 13. The monitor was a high-quality and color-calibrated Barco LC-47 at 1080p native resolution. The 30 subjects (24 male and 6 female of which 6 experts) were aged between 23 and 64 (mean 31).

Results are shown in the last column of Fig. 12. Confidence intervals are plotted according to the ITU-R BT.500-13 recommendation [52]. It is very interesting to notice that subjectively SMoE scores much better compared to the objective metrics PSNR and SSIM, with PSNR differences up to 6 dB. One explanation could be that the distortions introduced by SMoE (e.g. geometrical distortions and smoothing) are visually pleasing degradations. Furthermore, due to the continuous representation over all dimensions, SMoE is extremely view consistent. HEVC often introduced flickering when moving through views. We conclude that for MOS scores up to 4 (Perceptible but not annoying), we are competitive with motion-compensated HEVC. However, a MOS score of 5 (Imperceptible) remains hard to achieve as our kernels fail to capture higher spatial frequencies.

D. Light Field Video Coding

In our experiments, the RD-performance of MV-HEVC is compared to our SMoE approach. The MV-HEVC configuration includes only a single I-frame, each center-view per frame is predicted from other center-views. Other views in a frame are then subsequently estimated from the center-view. Fig. 14 illustrates the RD-curves in terms of SSIM (as SSIM correlated better with the subjective results above compared to PSNR). It is clear

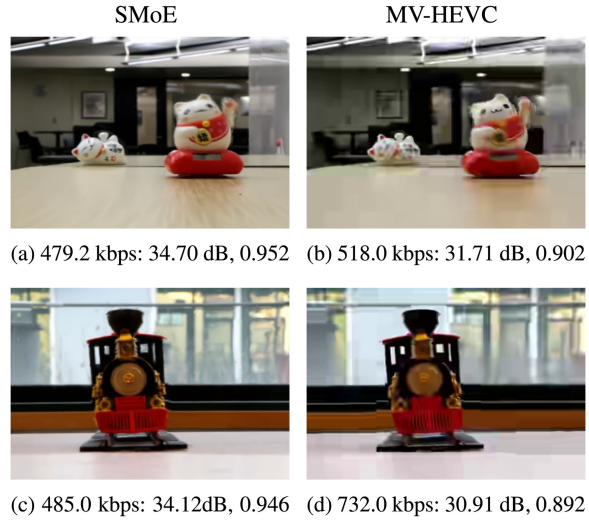


Fig. 15. Subjective comparison of SMoE (left) vs MV-HEVC (right) of *cats* (top) and *train1* (bottom) at frame 60. Metrics are indicated in PSNR and SSIM. SMoE is shown at respectively 7.5% and 33.7% less bitrate compared to MV-HEVC, while achieving superior objective quality around +3 dB PSNR and +0.05 SSIM.

that SMoE impressively outperforms MV-HEVC for the *cats* and *train1* sequences with bitrate savings up to a factor of 4x for the same quality. Table I shows the parameters used for the SMoE encoding of the model parameters and the corresponding results in PSNR and SSIM. Fig. 15 provides a subjective comparison. At low bitrates, SMoE results in spatio-temporal smoothing. In contrast, MV-HEVC typically exhibits more blocking artifacts, e.g. visible in the horizontal lines behind the train and around the train chimney. Furthermore, SMoE exhibits in general better temporal consistency, especially in static segments. Such kernels have a long spread along the temporal dimension t and the camera coordinate plane (a_1, a_2). As such, these kernels yield consistent views. In such areas, extra kernels can thus be used to increase spatial detail instead of temporal detail. To conclude, the application of the SMoE representation becomes increasingly interesting as the dimensionality p of the coordinate space increases.

VII. Conclusion and Future Work

This paper provided an in-depth presentation of Steered Mixture-of-Experts (SMoE): a novel framework for approximating image modalities with numerous applications. The image itself is not stored, instead, the underlying function that could have generated this image is approximated and stored. The function approximation is performed by dividing the coordinate space into many smaller patches that focus on parts of the underlying function (coinciding with regions in the image). Each such patch is approximated by a single expert-function, defined by a kernel. The total model thus consists of a set of kernels. The main benefits of SMoE compared to standard approaches are as follows. Firstly, the multi-dimensional kernels harvest long-term correlation over all dimensions simultaneously, which makes it inherently easy to tackle higher-dimensional image modalities. Secondly, although one Gaussian kernel has more parameters than e.g. a pixel, one kernel can cover thousands of pixels.

Thirdly, there is high potential for granular random access as each kernel is specialized in a region. Finally, the model provides a continuous approximation of the underlying pixel-generative function. Rendering a view thus means merely sampling that approximated function consisting only of the kernels relevant for that view. Furthermore, each pixel can be calculated independently, which thus allows for pixel-level parallel rendering.

In this work, we focused on the SMoE model for light field images and video and the application to coding. It was shown that the efficiency of SMoE increases when the dimensionality of the image modality increases. The reason is twofold. First, in dense representations the number of necessary pixels grows exponentially with the dimensionality. In contrast, sparse representations follow a more linear relationship depending on the image content. As a result, the average pixel-coverage by kernels increases exponentially as the dimensionality increases. Secondly, the number of parameters per kernel increases only linearly when the dimensionality increases. A dimension-agnostic coding scheme was introduced to binarize the kernel parameters. For static 4-D light fields, the SMoE-based codec was outperformed by HEVC when using motion-compensation (low random access, complex decoding structure) in terms of PSNR and SSIM, the SMoE-based codec did strongly outperform HEVC All-Intra (which allows similar granular random access as SMoE). Subjective tests were performed in order to assess view quality, view consistency, and refocusing after coding. These results remarkably show that SMoE is competitive with the best HEVC configuration up to the range of a MOS score above 4 (Perceptible but not annoying), arguably the most interesting range for coding schemes from a practical point of view. For 5-D light field video, we found that our approach can heavily outperform MV-HEVC up to bitrate savings up to a factor of 4x.

Our representation employs only linear regressors and thus assumes natural images to be able to be approximated as a smoothed piecewise linear function. However, the reality is more that image modalities resemble more piecewise stationary functions and can exhibit high spatial frequencies in textured regions. With the current model, an infeasible number of small kernels would be necessary to capture all detail. Current work aimed and succeeded in proving feasibility to design a sparse information-rich representation that scales to any dimensionality with desired functionality for VR consumption, e.g. random access, inherent view interpolation, and pixel-parallel reconstructions. Future work will thus consist of introducing provisions for residual texture. However, we have shown that even without these provisions, our model is competitive for low-to-mid range bitrates. Furthermore, the model is not trained to maximize PSNR of the reconstruction, but to maximize the likelihood of the model. As such, PSNR optimization could be a way to increase the RD-performance as early evidence shows [53], [54]. Finally, other properties and applications of SMoE need to be investigated and assessed.

Acknowledgment

The computational resources (STEVIN Supercomputer Infrastructure) and services used in this work were kindly provided by Ghent University, the Flemish Supercomputer Center (VSC), the Hercules Foundation, and the Flemish Government department EWI.

References

[1] I. Ihrke, J. Restrepo, and L. Mignard-Debise, "Principles of light field imaging: Briefly revisiting 25 years of research," *IEEE Signal Process. Mag.*, vol. 33, no. 5, pp. 59–69, Sep. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7559962/

[2] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. S. Landy and J. A. Movshon. Eds. Cambridge, MA, USA: The MIT Press, 1991, pp. 3–20.

[3] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. Conf. Comput. Graph. Interactive Techn.*, New York, NY, USA: ACM Press, 1996, pp. 31–42. [Online]. Available: http://portal.acm.org/citation.cfm?doid=237170.237199

[4] V. Avramelos *et al.*, "Highly parallel steered mixture-of-experts rendering at pixel-level for image and light field data," *J. Real-Time Image Process.*, Dec. 2018. [Online]. Available: http://link.springer.com/10.1007/s11554-018-0843-3

[5] 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 *Proc. IEEE Int. Conf. Image Process.*, Sep. 2016, pp. 2142–2146. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7532737http://ieeexplore.ieee.org/document/7532737/

[6] L. Lange, R. Verhack, and T. Sikora, "Video representation and coding using a sparse steered mixture-of-experts network," in *Proc. Picture Coding Symp.*, 2016, pp. 1–5.

[7] R. Verhack *et al.*, "Steered mixture-of-experts for light field coding, depth estimation, and processing," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2017, pp. 1183–1188. [Online]. Available: http://ieeexplore.ieee.org/document/8019442/

[8] I. Viola and T. Ebrahimi, "Quality assessment of compression solutions for ICIP 2017 grand challenge on light field image coding," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops*, Jul. 2018, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8551496/

[9] A. T. Hinds, D. Doyen, and P. Carballeira, "Toward the realization of six degrees-of-freedom with compressed light fields," in *Proc. IEEE Int. Conf. Multimedia Expo.*, Jul. 2017, pp. 1171–1176. [Online]. Available: http://ieeexplore.ieee.org/document/8019543/

[10] M. Domanski, O. Stankiewicz, K. Wegner, and T. Grajek, "Immersive visual media MPEG-I: 360 video, virtual navigation and beyond," in *Proc. Int. Conf. Syst., Signals Image Process.*, May 2017, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/7965623/

[11] G. Tech *et al.*, "Overview of the multiview and 3D extensions of high efficiency video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 35–49, Jan. 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7258339/

[12] J. P. Peixeiro, C. Brites, J. Ascenso, and F. Pereira, "Holographic data coding: Benchmarking and extending HEVC with adapted transforms," *IEEE Trans. Multimedia*, vol. 20, no. 2, pp. 282–297, Feb. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8013856/

[13] T. Georgiev, Z. Yu, A. Lumsdaine, and S. Goma, "Lytro camera technology: Theory, algorithms, performance analysis," C. G. M. Snoek, L. S. Kennedy, R. Creutzburg, D. Akopian, D. Wüller, K. J. Matherson, T. G. Georgiev, and A. Lumsdaine, Eds., Mar. 2013, p. 86671J. [Online]. Available: http://proceedings.spiedigitallibrary.org/proceeding.aspx?doi=10.1117/12.2013581

[14] Y. Li, M. Sjostrom, R. Olsson, and U. Jennehag, "Efficient intra prediction scheme for light field image compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2014, pp. 539–543. [Online]. Available: http://ieeexplore.ieee.org/document/6853654/

[15] L. F. R. Lucas *et al.*, "Locally linear embedding-based prediction for 3D holoscopic image coding using HEVC," in *Proc. 22nd Eur. Signal Process. Conf.*, 2014, pp. 11–15.

[16] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *Proc. IEEE Int. Conf. Multimedia Expo. Workshops*, Jul. 2016, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/7574671/

[17] I. Viola, M. Rerabek, and T. Ebrahimi, "Comparison and evaluation of light field image coding approaches," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1092–1106, Oct. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8010398/

[18] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo sequence based 2-D hierarchical coding structure for light-field image compression," in *Proc. Data Compression Conf.*, Apr. 2017, pp. 131–140. [Online]. Available: http://ieeexplore.ieee.org/document/7921908/

[19] C. Conti, L. Ducla Soares, and P. Nunes, "Light field coding with field of view scalability and exemplar-based inter-layer prediction," *IEEE Trans. Multimedia*, vol. 20, no. 11, pp. 2905–2920, Nov. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8338049/

[20] W. Ahmad, R. Olsson, and M. Sjostrom, "Interpreting plenoptic images as multi-view sequences for improved compression," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2017, pp. 4557–4561. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1134955&dswid=-1043http://ieeexplore.ieee.org/document/8297145/

[21] J. Chen, J. Hou, and L.-P. Chau, "Light field compression with disparity-guided sparse coding based on structural key views," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 314–324, Jan. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/8030107/

[22] B. Ceulemans, S. P. Lu, G. Lafruit, and A. Munteanu, "Robust multi-view synthesis for wide-baseline camera arrays," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2235–2248, Sep. 2018.

[23] T. Ebrahimi, S. Foessel, F. Pereira, and P. Schelkens, "JPEG Pleno: Toward an efficient representation of visual reality," *IEEE Multimedia*, vol. 23, no. 4, pp. 14–20, Oct.– Dec. 2016.

[24] M. B. de Carvalho *et al.*, "A 4D DCT-based lenslet light field codec," in *Proc. 25th IEEE Int. Conf. Image Process.*, Oct. 2018, pp. 435–439. [Online]. Available: https://ieeexplore.ieee.org/document/8451684/

[25] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Commun. Pure Appl. Math.*, vol. 42, no. 5, pp. 577–685, Jul. 1989. [Online]. Available: http://doi.wiley.com/10.1002/cpa.3160420503

[26] P. Prandoni and M. Vetterli, "Approximation and compression of piecewise smooth functions," *Philos. Trans. Roy. Soc. London. Series A: Math., Phys. Eng. Sci.*, vol. 357, no. 1760, pp. 2573–2591, Sep. 1999. [Online]. Available: http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.1999.0449

[27] H. Takeda, "Kernel regression for image processing and reconstruction," Ph.D. dissertation, Univ. California, Santa Cruz, CA, USA, 2006.

[28] D. S. Broomhead and D. Lowe, "Radial basis functions, multi-variable functional interpolation and adaptive networks," DTIC Doc., Tech. Rep. ADA196234, 1988. [Online]. Available: https://apps.dtic.mil/docs/citations/ADA196234

[29] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Stat. Comput.*, vol. 14, no. 3, pp. 199–222, 2004.

[30] R. Verhack, A. Krutz, P. Lambert, R. Van de Walle, and T. Sikora, "Lossy image coding in the pixel domain using a sparse steering kernel synthesis approach," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 4807–4811. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7025974

[31] O. Johannsen, A. Sulc, and B. Goldluecke, "Occlusion-aware depth estimation using sparse light field coding," in *Pattern Recognit. GCPR 2016. Lecture Notes in Computer Science*, B. Rosenhahn and B. Andres, Eds., Springer, Cham, 2016, pp. 207–218. http://link.springer.com/10.1007/978-3-319-45886-1_17

[32] O. Johannsen, A. Sulc, and B. Goldluecke, "What sparse light field coding reveals about scene structure," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3262–3270. [Online]. Available: http://ieeexplore.ieee.org/document/7780724/

[33] M. Hog, N. Sabater, and C. Guillemot, "Superrays for efficient light field processing," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1187–1199, Oct. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/8007186/

[34] S. Vagharshakyan, R. Bregovic, and A. Gotchev, "Light field reconstruction using shearlet transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 133–147, Jan. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7817742/

[35] D. S. Lalush and B. M. W. Tsui, "Block-iterative techniques for fast 4D reconstruction using a priori motion models in gated cardiac SPECT," *Phys. Medicine Biol.*, vol. 43, no. 4, pp. 875–886, Apr. 1998. [Online]. Available: http://stacks.iop.org/0031-9155/43/i=4/a=015?key=crossref.e338368ecee8b067ef47e68ae0c4e369

[36] S. E. Yuksel, J. N. Wilson, and P. D. Gader, "Twenty years of mixture of experts," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 8, pp. 1177–1193, Aug. 2012. [Online]: Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6215056http://ieeexplore.ieee.org/document/6215056/

[37] G. Bugmann, "Normalized gaussian radial basis function networks," *Neurocomputing*, vol. 20, no. 1–3, pp. 97–110, Aug. 1998. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0925231298000277

[38] H. Sung, "Gaussian mixture regression and classification," Ph.D. dissertation, Rice Univ., 2004. [Online]. Available: http://www.stat.rice.edu/%7B%25%7D7B%7B%7B %7D%7D%7B%25%7D7Dhgsung/thesis.pdf

[39] Z. Ghahramani and M. I. Jordan, "Supervised learning from incomplete data via an EM approach," in *Advances in Neural Information Processing Systems*, J. D. Cowan, G. Tesauro, and J. Alspector, Eds., San Mateo, CA, USA: Morgan-Kaufmann, 1994, pp. 120–127. [Online]. Available: http://papers.nips.cc/paper/767-supervised-learning-from-incomplete-data-via-an-em-approach.pdf

[40] T. Moon, "The expectation-maximization algorithm," *IEEE Signal Process. Mag.*, vol. 13, no. 6, pp. 47–60, Nov. 1996. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=543975

[41] M. Jordan and L. Xu, "Convergence results for the EM approach to mixtures of experts architectures," *Neural Netw.*, vol. 8, no. 9, pp. 1409–1431, Jan. 1995. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/0893608095000143

[42] P. Liang and D. Klein, "Online EM for unsupervised models," in *Proc. Human Lang. Technologies: Annu. Conf. North Amer. Chapter Assoc. Comput. Linguist.*, 2009, pp. 611–619.

[43] M.-A. Sato and S. Ishii, "On-line EM algorithm for the normalized gaussian network," *Neural Comput.*, vol. 12, no. 2, pp. 407–432, Feb. 2000. [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/089976600300015853

[44] D. G. Dansereau, "Light field toolbox for Matlab," 2015. [Online]. Available: https://nl.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4

[45] I. Viola *et al.*, "Objective and subjective evaluation of light field image compression algorithms," in *Proc. 32nd Picture Coding Symp.*, 2016, Paper EPFL-CONF-221601.

[46] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. Qual. Multimedia Experience*, 2016, Paper EPFL-CONF-218363.

[47] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004. [Online]. Available: http://ieeexplore.ieee.org/document/1284395/

[48] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Series B (Methodological)*, vol. 39, pp. 1–38, 1977.

[49] D. Sculley, "Web-scale k-means clustering," in *Proc. 19th Int. Conf. World Wide Web*. New York, NY, USA: ACM Press, 2010, p. 1177. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1772690.1772862

[50] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–13, Jul. 2017. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3072959.3073614

[51] K. McCann *et al.*, "High efficiency video coding (HEVC) test model 16 (HM 16) improved encoder description," ITU-T Joint Collaborative Team on Video Coding (JCT-VC), Sapporo, Japan, Tech. Rep. JCTVC-S1002, 2014. [Online]. Available: https://phenix.int-evry.fr/jct/doc_end_user/current_document.php?id=9468

[52] ITU-R, "Recommendation ITU-R BT.500-13," Int. Telecommun. Union, Tech. Rep., 2012. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/bt/r-rec-bt.500-13-201201-i!!pdf-e.pdf

[53] M. Tok, R. Jongebloed, L. Lange, E. Bochinski, and T. Sikora, "An MSE approach for training and coding steered mixtures of experts," in *Proc. IEEE Picture Coding Symp.*, San Francisco, CA, USA, Jun. 2018, pp. 273–277. [Online]. Available: https://ieeexplore.ieee.org/document/8456250/

[54] E. Bochinski, R. Jongebloed, M. Tok, and T. Sikora, "Regularized gradient descent training of steered mixture of experts for sparse image representation," in *Proc. 25th IEEE Int. Conf. Image Process.*, Athens, Greece: IEEE, Oct. 2018, pp. 3873–3877. [Online]. Available: https://ieeexplore.ieee.org/document/8451823/