Light Field Image Compression Using Generative Adversarial Network-Based View Synthesis

Chuanmin Jia, *Student Member, IEEE*, Xinfeng Zhang, *Member, IEEE*, Shanshe Wang, Shiqi Wang, *Member, IEEE*, and Siwei Ma, *Senior Member, IEEE*

***Abstract*— Light field (LF) has become an attractive representation of immersive multimedia content for simultaneously capturing both the spatial and angular information of the light rays. In this paper, we present a LF image compression framework driven by a generative adversarial network (GAN)-based sub-aperture image (SAI) generation and a cascaded hierarchical coding structure. Specifically, we sparsely sample the SAIs in LF and propose the GAN of LF (LF-GAN) to generate the unsampled SAIs by analogy with adversarial learning conditioned on its surrounding contexts. In particular, the LF-GAN learns to interpret both the angular and spatial context of the LF structure and, meanwhile, generates intermediate hypothesis for the unsampled SAIs in a certain position. Subsequently, the sampled SAIs and the residues of the generated-unsampled SAIs are re-organized as pseudo-sequences and compressed by standard video codecs. Finally, the hierarchical coding structure is adopted for the sampled SAI to effectively remove the inter-view redundancies. During the training process of LF-GAN, the pixel-wise Euclidean loss and the adversarial loss are chosen as the optimization objective, such that sharp textures with less blurring in details can be produced. Extensive experimental results show that the proposed LF-GAN-based LF image compression framework outperforms the state-of-the-art learning-based LF image compression approach with on average 4.9% BD-rate reductions over multiple LF datasets.**

***Index Terms*— Light field image compression, SAI synthesis, adversarial learning, hierarchical coding.**

I. Introduction

LIGHT fields contain rich representations of real-world objects and scenes, which enable tremendous applications such as view synthesis, depth estimation and 3D reconstruction. The commercialized light field cameras (e.g., Lytro [1] and RayTrix [2]) provide flexibility for consumers with free viewpoint change and post-processing for photograph refocusing. Hence, light field can also be a solution for immersive multimedia applications such as 3D gaming and movies, etc. Basically, two kinds of data formations can be utilized for LF visualization, sub-aperture image (SAI, Fig. 1(a)) representation and epipolar-plane image (EPI, Fig. 1(b)) representation, which are illustrated in Fig. 1. Inspired by plenoptic function [4], the SAIs originate from LF structure parameterized by four elements (x, y, u, v), as illustrated in Fig. 2(a), in which (x, y) indicates the lenslet angular geometry and (u, v) denotes the spatial position. However, the EPIs can be constructed by re-sampling one single row/column and restricting one spatial coordinate and one directional coordinate as the constant. The details of SAI generation and EPI generation are shown in Fig. 2(b) and Fig. 1(b), respectively.

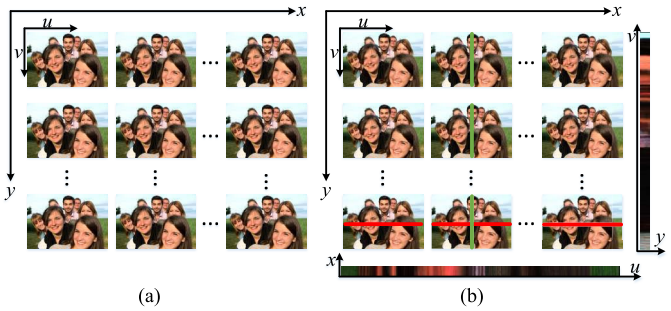


Fig. 1. Different formations of LF [3]: (a) SAI representation (b) EPI representation. Bottom EPI: sampled from red lines; right EPI: sampled from green lines.

The LF imaging process captures rich information, not only in terms of the intensity of scenes but also the directions of the light rays, which result in highly redundant data. This further poses great challenges to the transmission bandwidth, storage and processing (read/write). For instance, the raw LF image captured by Lytro camera contains $5368 \times 7728 \times 3$ pixels which require over 20 times more memory for storage than a single full high-definition (HD) image [6]. Moreover, after decomposing LF into 4-D structure, the angular resolution of Lytro camera captured LF image is 15×15 and the spatial resolution is 434×625, which indicates that there are $15 \times 15 \times 434 \times 625 \times 3$ pixels from both spatial and

Manuscript received July 22, 2018; revised October 8, 2018; accepted December 5, 2018. Date of publication December 13, 2018; date of current version March 11, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant 61872400, in part by the Top-Notch Young Talents Program of China, in part by the high-performance computing platform of Peking University, in part by the Hong Kong RGC Early Career Scheme under Grant 9048122 (CityU 21211018), in part by the City University of Hong Kong under Grant 7200539/CS, and in part by the China Scholarship Council under Grant 201706010248. This paper was recommended by Guest Editor W.-H. Peng. *(Corresponding author: Siwei Ma.)*

C. Jia, S. Wang, and S. Ma are with the Institute of Digital Media, Peking University, Beijing 100871, China (e-mail: cmjia@pku.edu.cn; sswang@pku.edu.cn; swma@pku.edu.cn).

X. Zhang is with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA 90089 USA (e-mail: zhangxinf07@gmail.com).

S. Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: shiqwang@cityu.edu.hk).

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/JETCAS.2018.2886642

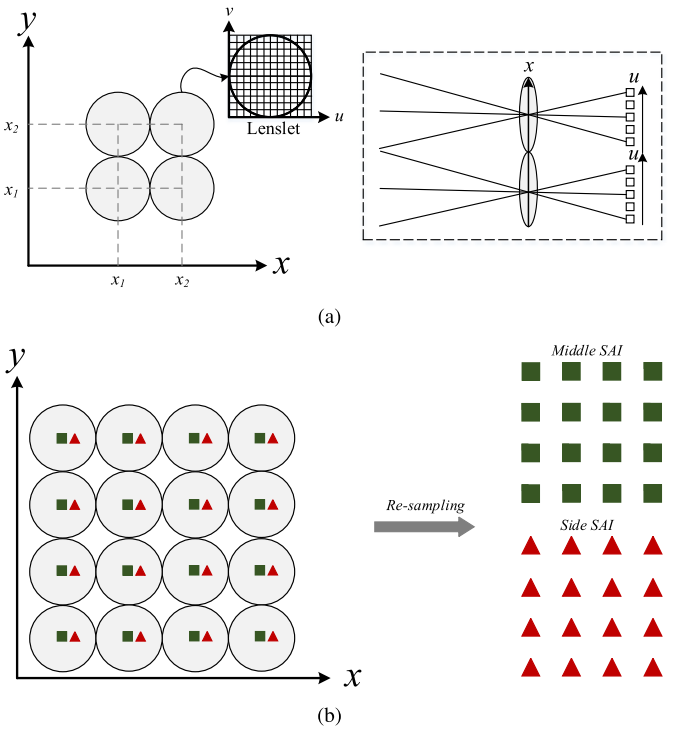


Fig. 2. Illustrating the process from lenslet to LF-4D structure: (a) LF-4D parameterizations [5]. LF can be parameterized by the lenslet positions (x, y) and the pixel positions (u, v) behind one lenslet; (b) Generation of SAIs by re-sampling pixels in lenslet.

angular domains. Hence, there is a strong demand for the high efficiency compression methods for LF images.

Recently, learning based algorithms for LF image processing and compression have achieved promising results [3], [6]–[8]. Tremendous attentions have been focused on convolutional neural network (CNN) based approaches for spatial and angular super-resolution (SR) of LF-4D [3], [9]–[11], which inspire us to investigate the synthesis and compression approach of LF content from the learning perspective. In this paper, we propose a LF image compression framework by taking the advantage of generative adversarial network (GAN). More specifically, the LF SAIs are firstly sparse-sampled to obtain the pseudo-sequence which is then compressed by a standard video codec. Subsequently, the unsampled SAIs are generated by the proposed LF generative adversarial network (LF-GAN) which involves the sampled-then-compressed SAIs as priors. To achieve high efficiency LF image compression, the residue SAIs between the generated SAIs and their original signals are also re-arranged as a pseudo-sequence to enhance the coding efficiency. Moreover, since the particular geometrical structure between SAIs makes the pseudo-sequence essentially different from natural videos, we adopt a cascaded hierarchical coding structure for the standard video codec to efficiently exploit the inter-view correlations. Regarding the proposed LF-GAN, we utilize the conditional adversarial learning approach to implicitly adapt the scene of disparity in different perspectives. More specifically, the sampled SAIs are modeled as the LF context prior for the unsampled intermediate views. The main contributions of this paper can be summarized as follows:

- We propose a generative adversarial learning network for LF (LF-GAN) SAI generation which, to the best of our knowledge, is the first algorithm utilizing GAN model for LF image compression.
- We establish the cascaded hierarchical coding structure to facilitate the generation of SAIs as well as the optimal bit allocation scheme for pseudo-sequences formed by sparsely-sampled SAIs.
- The proposed LF image coding framework outperforms the state-of-the-art learning-based LF image compression approach in terms of rate-distortion (RD) performance.

The remainder of this paper is organized as follows. Section II reviews the related work. In Section III, we present details of the proposed LF-GAN approach for LF SAI generation. Section IV introduces the proposed LF image coding framework using the proposed LF-GAN. Section V presents the details of our implementation. We then validate the performance of LF image compression in Section VI and more analyses are also provided. Finally, the conclusion and future work are discussed in Section VII.

II. Related Work

In this section, we briefly revisit the related literatures from the following aspects: LF view synthesis, LF compression and deep learning (DL) based LF processing.

A. LF View Synthesis

One possible solution for LF view synthesis is the image based rendering (IBR) techniques [12], the philosophy of which focuses on synthesizing the intermediate views via depth estimation from disparity maps between stereo views [13]–[15]. However, these kind of methods often suffer from severe visual artifacts due to simply averaging the warped neighboring views after depth estimation for each input image. Another category for efficient LF synthesis is based on the concept of plenoptic function [4], [5]. In particular, the LF capturing process and the intermediate view generation problem can be regarded as sparse-sampling and approximation of the original plenoptic function. Specifically, the pixels of the captured views are treated as sampling results of a multidimensional LF function, such that the unknown intermediate views can be formulated as function values (i.e., $L_{unknow} = f(L_{sampled})$, f denotes the function which defines the relationship between the sampled views and the unknow view) determined after its reconstruction from sampled results. As such, the synthesis of missing view in LF can be solved by using interpolation methods [16], [17]. The EPI based methods are also explored to synthesize the desired perspective views within the LF structure [18]. The sparse representation of the EPIs in transform domain [19], [20] has also been investigated for intermediate view synthesis.

B. LF Compression

Numerous LF image compression methods have been recently proposed [21], [22] and the standardization process

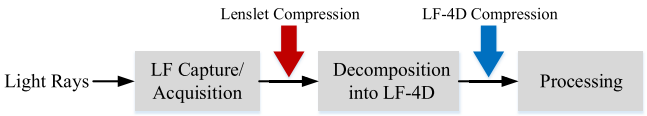


Fig. 3. The diagram of acquisition, compression and processing pipeline for LF content.

has also been initiated by JPEG Pleno [23]. There are mainly two kinds of frameworks for LF coding, and the illustrations for such two categories of LF image compression methods are depicted in Fig. 3. The first kind of coding approach directly compresses the raw LF sensor data obtained after the LF acquisition step (red arrow in Fig. 3) and the lenslet based intra coding algorithms are proposed by investigating non-local self-similarity compensation prediction and local linear embeddings [24]–[27]. This kind of approaches usually integrates the philosophy of multi-hypothesis prediction [28] into intra LF image compression and achieves significant performance improvement in terms of coding gain and subjective quality for reconstructed LF 4-D structure [29], [30]. Moreover, for plenoptic contents, the sensor-adaptive transform and reshaping methods were proposed in [31] and [32] to efficiently compress the lenslet images. The other category considers the 4D representations of LF for compression (blue arrow in Fig. 3). As such, the pseudo-sequence based algorithms [8], [33]–[38] have been designed to decompose the original lenslet data into SAIs. Subsequently, the SAIs are organized into pseudo-sequences and compressed by video encoder to reduce both the intra- and inter-frame redundancy. Inter prediction can be investigated to adapt the content variations for LF compression. These categories of LF coding methods usually focus on the coding structure design of SAIs [35]–[37], bit-rate allocation with rate-distortion optimization (RDO) [8], [36] and sparse coding [38], [39] for intermediate SAIs. Moreover, various scan orders during pseudo-sequences generation from SAIs are explored, such as zig-zag, spiral, raster, Hilbert, rotation, and hybrid of horizontal zig-zag and U-shape. Extensive experiments have been conducted to evaluate their performance in [35]and [40]–[42]. In our previous work [43], we proposed a CNN based coding scheme with the jointly optimized post-processing networks for LF image compression.

C. DL Based LF Processing

DL based approaches have also achieved significant advances in LF image processing, such as CNN based view synthesis [7], LF reconstruction [3], LF super-resolution [44], LF video caption [6], depth estimation [45]. Kalantari *et al.* [7] proposed CNN models for disparity estimation and color prediction to synthesize views in LF images, which can mitigate the trade-off between the angular and spatial resolution in consumer light field cameras. Gul and Gunturk [9] established the light field super resolution (LFSR) framework to enhance both of the spatial and angular resolution for LF-4D structure. As for LF video imaging, Wang *et al.* [6] investigated the hybrid imaging system based on LF camera and conventional camera to obtain LF videos, which could achieve 30 frame-per-second (fps) LF video caption with the help of CNN models. Yoon *et al.* [44] first augment the spatial resolution of each SAI to enhance details by a spatial SR network. Then, the unsampled views between the existing SAIs are generated by an angular super-resolution CNN network. To achieve the single view depth estimation, Garg *et al.* [45] proposed an unsupervised framework to learn CNN model for single view depth prediction, without requiring annotated ground-truth depths.

III. LF Generative Adversarial Networks

In this section, we introduce the LF-GAN which is adopted in the compression framework. Specifically, we will first introduce the whole network architecture of LF-GAN which mainly contains three core components to achieve high quality LF SAI generation. First, by taking the the surrounding sampled SAIs as input, the multi-branch fusion network assimilates the angular context as well as the spatial information within SAIs to generate the high order approximation of the target perspective view of SAI. The advantage of the proposed multi-branch fusion network lies in that no explicit requirement for depth information from the input SAIs when comparing with IBR based algorithms. Subsequently, to better enhance the image quality of generated high-order approximation SAI from signal-preserving perspective, the refinement-generator network is then applied after the fusion network. The refinement-generative network aims at directly mapping the high order approximation from fusion network to the generated high quality SAI (SAI_{refine}). Finally, with the guidance and assistance of adversarial learning [46] philosophy, we propose the discriminative network to sharpen the edges and improve the subjective quality of output SAIs generated by previous two sub-networks. The illustrations of entire learning architecture are provided in Fig. 4, including the overall infrastructure of LF-GAN as well as each component, the multi-branch fusion network, refinement generative network and discriminator.

A. Multi-Branch Fusion Network

To obtain the intermediate missing views (SAIs), the context information from both angular and spatial domains should be taken into consideration. Here, we firstly design a multi-branch fusion network (MBFN) to acquire a high order approximation of the desired SAIs. As shown in Fig. 4, MBFN utilizes p-neighbor (p equals the number of branches) SAIs as inputs for the multi-branches, each of which shares the same network structure design but with different $weights$ and $bias$. Each neighboring SAI is then fed into one individual branch consisting of four convolution layers with non-linear activations, and then the high order approximation of the target SAI can be generated via concatenation and fusion for $p-$ branches. The normative definition of high order approximation can be organized as follows,

$$SAI_{approx} = \mathbf{F}_{MBFN}(SAI_1, SAI_2, \ldots, SAI_p), \qquad (1)$$

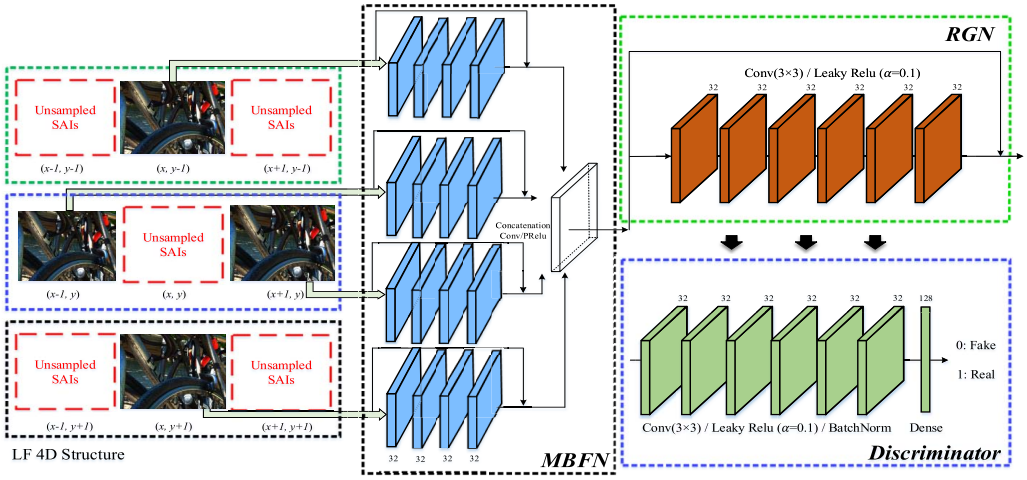


Fig. 4. The detail architecture of proposed LF-GAN framework. The fusion network first obtains the approximation of the targeting SAI. Subsequently, the refinement generative network generates the final estimation. To ensure sharp edges as well as detailed textures, the adversarial training is also utilized during the final generation process.

where $SAI_i, i = \{1, \ldots, p\}$ are input perspective SAI images and $\mathbf{F}_{MBFN}(\cdot)$ denotes the proposed MBFN, each branch of which is parameterized by

$$L_j(0) = SAI_j, \tag{2}$$

$$L_j(i) = PReLU(W_{ji} \times L_j(i-1) + b_{ji}), \quad i = 1, 2, \ldots, k-1, \tag{3}$$

$$L_j(k) = W_{jk} \times L_j(k-1) + b_{jk} + SAI_j. \tag{4}$$

Here j is the index and indicates the j-th branch of MBFN which contains k convolution layers, and $L_j(i)$ is i-th layer of j-th branch of MBFN. It is worth noting that the Parametric Rectified Linear Unit (PReLU) nonlinearity [47] is adopted as our activation function for the first $(k-1)$-th convolution layers, Meanwhile, the residue connection is also deployed to speed-up learning process which is described in Eqn (4). It should be also noted that we further utilized 1×1 convolution for the concatenated feature maps of $p-$branches such that the output of MBFN is converted into image domain with one channel.

$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0, \\ \dfrac{x_i}{a_i} & \text{if } x_i < 0, \end{cases} \tag{5}$$

where a_i is a learnable parameter for each convolution output channel x_i in the training process via back propagation. Hence, the learning loss function of MBFN is formulated in Eqn (6).

$$\mathcal{L}_{MBFN}(SAI_{target}, SAI_{approx}|\eta) = \sum_{i=0}^{wid-1} \sum_{j=0}^{hgt-1} \times (SAI_{target}[i][j] - SAI_{approx}[i][j])^2 + \lambda \|\eta\|^2, \tag{6}$$

where η capsules the network parameters and SAI_{target} denotes the label SAI. Moreover, the network parameter settings are described in Appendix.

B. Refinement Generative Network

After approximating the target SAI via MBFN by taking advantages of the neighboring SAIs, the refinement generative network (RGN) is then proposed to enhance the generated SAI quality. By building the bridge between the label SAI domain and high order approximation domain, RGN tends to directly learn the mapping between such two spaces with the objective of minimizing the L_2 distance between SAI_{target} and $RGN(SAI\_approx)$,

$$\mathcal{L}_{RGN}(SAI_{target}, SAI_{approx}|\Theta) = \sum_{i=0}^{wid-1} \sum_{j=0}^{hgt-1} (SAI_{target}[i][j] - RGN(SAI_{approx}[i][j]))^2 + \lambda \|\Theta\|^2, \tag{7}$$

where wid and hgt are the width and height of each SAI, the L_2 norm of Θ is the regularizer for the parameters of RGN.

In analogous to MBFN, we use fully convolutional layers for the network structure of RGN which takes the SAI_{approx} as input and generates the $\text{SAI}_{refine}(SAI_{refine} = RGN(SAI_{approx}))$. The loss function in Eqn (7) is adopted for optimization. RGN contains six convolution layers, and the first five layers of RGN adopt PReLU activation after convolution operation. By contrast, the last layer of RGN does not contain any non-linearity function. Moreover, the intermediate layers also apply batch normalization after activation to speed-up the training process [48]. Table III in Appendix provides the details of each layer in RGN.

C. Discriminative Network

With the great success achieved by adversarial learning [46], the unsupervised adversarial learning is widely adopted and utilized in image generation [49], image-to-image translation tasks [50] and generative compression for images [51]. Since disparity exists between different SAIs,

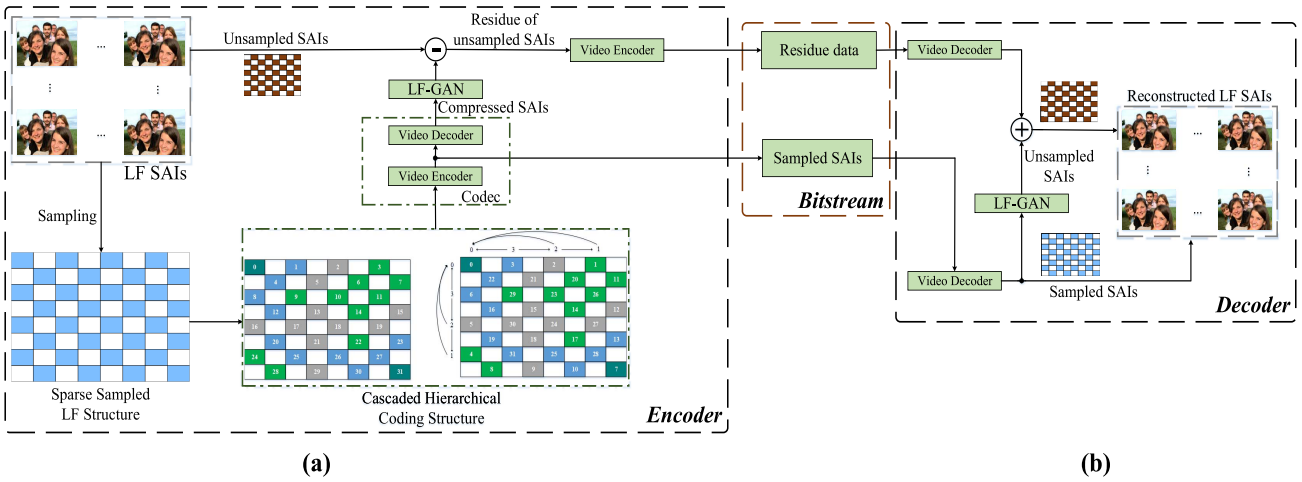


Fig. 5. The diagram of proposed LF image coding framework via GAN based view synthesis. (a) Encoder; (b) Decoder.

the trivial convolution layers cannot handle such external difference among SAIs. To tackle this issue, we establish the learning framework with heuristic generative-adversarial module which is able to compensate the SAI disparity from the unsupervised learning perspective. In particular, the previous RGN and one discriminator D_η, parametrized η, are jointly optimized with following learning objective:

$$\min_{\theta}\max_{\eta}\mathbb{E}_{y\sim p_{target}}\log D_\eta(y) + \mathbb{E}_{x\sim p_{refine}}\log(1-D_\eta(RGN(x))), \quad (8)$$

where p_{target} and p_{refine} are the empirical distributions of SAI_{target} and SAI_{refine} training samples, respectively.

The architecture of D_η is shown in Fig. 4, which is composed of six convolution layers followed by two dense connected layers with dimension 128 and 1, respectively. The network structure is listed in Table IV. Basically, the number of feature maps is identical from the first layer to the sixth convolution layer. The receptive field of each convolution layer is restricted into 3×3 with *same* padding strategy, and two fully connected layers are responsible for the dimension reduction from feature space to the discriminative results. Hence, the objective of D_η is to generate a binary decision for the sake of discriminating whether the output of RGN's output is $real$ or $fake$. In this manner, the minor disparity can be compensated for the generated SAIs, and sharper edges as well as realistic textures can also be obtained.

D. Loss Function Designation

To guarantee the high quality signal restoration as well as the perceptual quality for LF-GAN, both adversarial loss [46] and mean square loss are adopted in our final model. The loss for MBFN, RGN as well as the discriminator D are simultaneously optimized by assigning hyper-parameters for them to achieve multi-task learning,

$$\mathcal{L}=\lambda_1*\mathcal{L}_{MBFN}+\lambda_2*\mathcal{L}_{RGN}+\lambda_3*\mathcal{L}_D,\quad \sum_i\lambda_i=1, \quad (9)$$

where the $\mathcal{L}_{MBFN}$ and $\mathcal{L}_{RGN}$ denote the mean square loss of MBFN and RGN respectively, $\mathcal{L}_D$ is the adversarial loss. It is worth noting that the hyper-parameters λ_i are empirically set during implementation. Herein, the adversarial loss $\mathcal{L}_D$ acts the role of encouraging RGN to favor the generated results in the manifold of SAIs,

$$\mathcal{L}_D=\sum_i\log(1-D_\eta(RGN(x_i))). \quad (10)$$

IV. LF Image Compression Framework With GAN Based View Synthesis

The proposed LF image compression framework with GAN based view synthesis is presented in this section. We first demonstrate the overall framework of the proposed coding framework for LF images. Subsequently, we introduce the details of the proposed cascaded hierarchical coding structure for pseu do-sequence based SAI coding. Finally, RD analysis is performed to achieve optimal bit allocation in the encoding process.

A. Overview

The entire working flow of the proposed LF image compression framework is depicted in Fig. 5. For the encoder side, the $N\times N$ ($Nequals8$ in Fig. 8(b)) SAIs are first sparsely sampled and then re-organized into pseudo-sequence according to the raster scan order. The red arrow in Fig. 8(b) illustrates the procedure from SAIs to pseudo-sequence. It should be noted that the blue rectangles represent the sampled SAIs while white ones are unsampled SAIs. Once the pseudo-sequences are obtained, the standard video codec is utilized to remove the intra- and inter-SAI redundancy. Since the SAI based pseudo-sequence has different content geometry features and RD characteristics, we propose the cascaded hierarchical coding structure to establish the reference frame relationships for the coding of pseudo-sequences. Moreover, the optimal bit allocation scheme for such coding structure is also explored. To generate the unsampled SAIs, the proposed LF-GAN takes the sampled-and-compressed SAIs as LF context prior and synthesizes the target SAIs after joint training of MBFN,

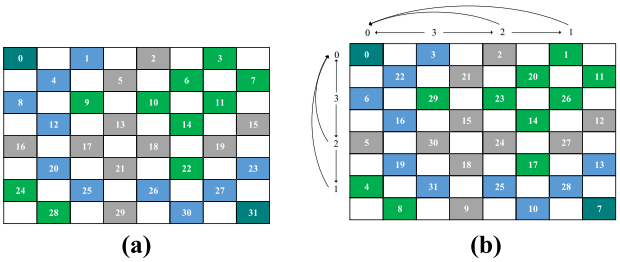


Fig. 6. The proposed cascaded hierarchical coding structure, where the white rectangles are un-sampled SAIs. (a) picture order count (POC) for raster scan; (b) the proposed coding order. There are four different temporal layers for SAIs (dark green: layer-0, green: layer-1, gray: layer-2, blue: layer-3).

RGN and discriminator. Since the input of LF-GAN is the codec-compressed SAIs, the compression error will propagate to the synthesized SAIs through the networks. To address this issue and further enhance the quality of synthesized SAIs, the residue data of unsampled SAIs are obtained by subtraction between original signals and LF-GAN synthesized signals at the encoder side. After re-arranging the residue SAIs into pseudo-sequence as previously mentioned method, the residue data can also be compressed using standard video codec and transmitted to the decoder. Hence, the bitstream of proposed LF image coding framework consists of two separate segments, the bitstream for sampled SAIs and the bitstream for residue data of unsampled SAIs. In general, compared to [8], our framework offers more flexibility to intermediate view synthesis due to GAN based model. Moreover, different from one model for all circumstances in [8], particular models are trained for different QP values to adapt to different distortion level.

B. Coding Structure

The cascaded hierarchical coding structure which treats the whole pseudo-sequence as a group-of-pictures (GOP) is employed [37], as shown in Fig. 6. Such reference picture order can facilitate the quality of generated unsampled SAIs since the coding order is a two-dimensional extension of conventional hierarchical reference scheme in HEVC, which is able to provide the high-quality reference pictures during coding. It is worth noting that there are four different temporal layers for SAIs in our coding structure, as illustrated in Fig. 6(b), in which different colors correspond to different layers in Fig. 6. The dark green rectangles correspond to layer-0, green rectangles indicate layer-1, gray rectangles mean layer-2 and blue rectangles represent layer-3. The pictures with higher layer numbers can select the lower layers pictures as references. Inspired by the work in [37], the proposed coding structure utilizes first-row-then-column temporal layer based coding order, which implies horizontal SAIs are coded first and then vertical SAIs. Taking picture order count (POC) $20, 21, 22, 23$ in Fig. 6(a) as an example, the coding order for this row should be $19, 18, 17, 13$ in Fig. 6(b). As such, the number of reference pictures for frame POC-19 is 10, which is also the maximum size of reference picture buffer in our adopted coding structure.

C. Joint RD Optimization Bit Allocation

Due to the fact that the residue data is treated as the source signal for compression, the pseudo-sequences hold completely different RD characteristics compared with conventional natural videos. As such, existing bit allocation algorithm in the standard video codec cannot be directly applied for the LF image compression. Hence, we propose the joint R-D optimal bit allocation mechanism based on the LF coding structure to achieve a good balance between sampled SAIs and unsampled SAIs representation. The optimization objective of our RDO is to minimize the distortion D of compressed pseudo-sequence with the given bit budget R_t. As proposed in [8] and [52], the analysis methods for RD model are investigated via rate-quantization (R-Q) model and distortion-quantization (D-Q) model. Therefore, the RD models for sampled SAIs pseudo-sequence and unsampled SAIs pseudo-sequence encoding are jointly formulated as follows,

$$\min_{q_{i,sp},q_{i,usp}} \frac{1}{2\times K}\sum_{i=1}^{K}[D_{i,sp}(q_{i,sp})+D_{i,usp}(q_{i,sp},q_{i,usp})],$$
$$\text{s.t.}\sum_{i=1}^{K}[R_{i,sp}(q_{i,sp})+R_{i,usp}(q_{i,sp},q_{i,usp})]\le R_t, \quad (11)$$

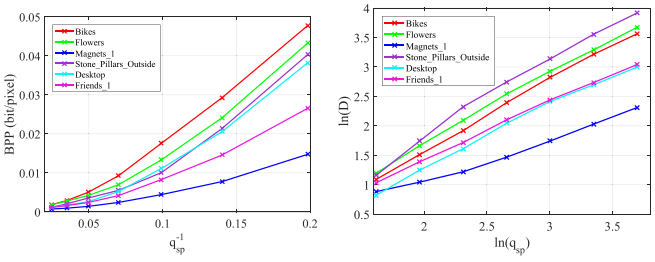
where $q_{i,sp}$ and $q_{i,usp}$ are the quantization step for sampled SAIs and unsampled SAIs respectively, $D_{i,sp}$ and $R_{i,sp}$ represent the distortion and rate for the i-th frame and K denotes the number of frames in the sampled SAI pseudo-sequence. Analogously, $D_{i,usp}$ and $R_{i,usp}$ represent the distortion and rate for the residue of unsampled SAI pseudo-sequence. It should be noted that K equals 32 for the sampled-SAI-pseudo-sequence and the residue of unsampled-SAIs-pseudo-sequence. Moreover, each frame corresponding to the unsampled SAIs[1] can be regarded as the residue images. After incorporating the Lagrangian multipliers into Eqn (11), the constrained optimization problem can be converted into the following unconstrained one:

$$\min_{q_{i,sp},q_{i,usp},\lambda} \frac{1}{2\times K}\sum_{i=1}^{K}[D_{i,sp}(q_{i,sp})+D_{i,usp}(q_{i,sp},q_{i,usp})]$$
$$+\lambda\sum_{i=1}^{K}[R_{i,sp}(q_{i,sp})+R_{i,usp}(q_{i,sp},q_{i,usp})-R_t], \quad (12)$$

where the λ is Lagrange Multiplier of the optimization problem. To solve the optimization problem, we establish the D-Q model and R-Q model for sampled SAIs and unsampled SAIs. It is worth mentioning that similar studies have been conducted in [8] and [52], and our modelling is particularly derived for the GAN based view synthesis, since the views generated by LF-GAN have different RD behaviour.

1) RD Model for Sampled SAIs: For sampled SAIs, Fig. 7(a) plots actual encoding bits with different $q_{i,sp}$ values for four LF images, where we can observe that there exists exponential relationship between the rate (bit-per-pixel, BPP) and the inverse of $q_{i,sp}$, which is different from the linear

[1]For brief discussion, we denote *the residue of unsampled-SAIs-pseudo-sequence* as *unsampled SAIs*, *sampled-SAIs-seudo-sequence* as *sampled SAIs*.



(a) (b)

Fig. 7. (a) Illustration of the rate-quantization relation; (b) illustration of the distortion-quantization relation with different Qstep values ($q_{i,sp}$) for the six LF images.

R-Q model in [8]. As for the D-Q model, an obvious linear relationship between the distortion (mean-square-error, MSE) and $q_{i,sp}$ in a log-scale can also be observed according to Fig. 7(b).

Therefore, we use the natural exponential function and power function in terms of $q_{i,sp}$ to model the R-Q and D-Q model of the sampled SAIs, respectively,

$$R_{i,sp}(q_{i,sp}) = a * e^{q_{i,sp}^{b}}, \tag{13}$$

$$D_{i,sp}(q_{i,sp}) = \theta * q_{i,sp}^{\phi}, \tag{14}$$

where a, b, θ and ϕ are the empirical parameters fitted by the two-pass encoding scheme which will be described in the next Section.

2) RD Model for Unsampled SAIs: For unsampled SAIs, the modelling is complicated as the prediction of unsampled SAIs is generated via compressed sampled SAIs using LF-GAN. Then the residue data are obtained by subtracting the original signal with the generated prediction value. Therefore, there are two major factors which have influence on the RD model of residue data: $q_{i,usp}$ and $q_{i,sp}$. For simplicity, we directly reuse the model proposed by *Hou et al.* in [8], where the R-Q model for unsampled SAIs is defined as,

$$R_{i,usp}(q_{i,usp}, q_{i,sp}) = c * q_{i,sp}^{\beta} * q_{i,usp}^{\gamma}, \tag{15}$$

where $c > 0, \beta > 0$ and $\gamma > 0$ are model parameters. The D-Q model for unsampled SAIs are formulated as follows,

$$D_{i,usp}(q_{i,usp}, q_{i,sp}) = \theta * q_{i,sp}^{\phi} + (p_1 q_{i,sp}^{\tau} + p_0) q_{i,usp}^{-1} + (k_1 q_{i,sp}^{\mu} + k_0). \tag{16}$$

By eliminating the intermediate variables, we are able to solve the optimization problem in Eqn (11) via substituting the terms with Eqn (13)–(16) to Eqn (12). Hence, the unconstrained optimization problem is given by,

$$\min_{q_{i,sp}, q_{i,usp}, \lambda} J = \frac{1}{K} \sum_{i=1}^{K} [\theta * q_{i,sp}^{\phi} + \frac{1}{2}(p_1 q_{i,sp}^{\tau} + p_0) q_{i,usp}^{-1} + \frac{1}{2}(k_1 q_{i,sp}^{\mu} + k_0)] + \lambda \sum_{i=1}^{K} [a e^{q_{i,sp}^{b}} + c q_{i,sp}^{\beta} * q_{i,usp}^{\gamma} - R_t]. \tag{17}$$

However, Eqn (17) is an underdetermined and non-convex problem. To tackle this issue, we introduce the equal-QP assumption to simplify the equation. Therefore, the following condition can be imposed,

$$q_{i,sp} = q_{sp}, q_{i,usp} = q_{usp}, \quad i = 1, \ldots, K. \tag{18}$$

It should be noted that Eqn (18) implies that each frame in sampled SAIs uses identical QP value. For the unsampled SAIs, we have the same constraint. Therefore, the Eqn (17) can be elegantly re-written as,

$$\min_{q_{sp}, q_{usp}, \lambda} J = [\theta q_{sp}^{\phi} + \frac{1}{2}(p_1 q_{sp}^{\tau} + p_0) q_{usp}^{-1} + \frac{1}{2}(k_1 q_{sp}^{\mu} + k_0)] + \lambda (a e^{q_{sp}^{b}} + c q_{sp}^{\beta} q_{usp}^{\gamma} - R_t). \tag{19}$$

For the above function J, we take the first-order partial derivative respect to λ, q_{sp}, q_{usp} to obtain $\frac{\partial J}{\partial \lambda}$, $\frac{\partial J}{\partial q_{sp}}$, and $\frac{\partial J}{\partial q_{usp}}$ respectively. After setting them to zeros, we could obtain the following relationship,

$$\begin{cases} a e^{q_{sp}^{b}} + c q_{sp}^{\beta} q_{usp}^{\gamma} = R_t \\ (\phi \theta q_{sp}^{\phi-1} + \frac{\tau}{2} p_1 q_{sp}^{\tau-1} q_{usp}^{-1} + \frac{\mu}{2} k_1 q_{\mu-1}^{sp}) \\ \quad + \lambda (a b e^{q_{sp}^{b}} q_{sp}^{b-1} + c\beta q_{sp}^{\beta-1} q_{usp}^{\gamma}) = 0 \\ \quad - \frac{1}{2}(p_1 q_{sp}^{\tau} + p_0) q_{usp}^{-2} + \lambda c \gamma q_{sp}^{\beta} q_{usp}^{\gamma-1} = 0. \end{cases} \tag{20}$$

Therefore, once all the parameters in Eqn (20) are determined, the optimal bit allocation scheme can be achieved by solving the q_{sp} and q_{usp}.

V. Implementation

The implementation details of the proposed coding framework are presented in this section, including the network architecture, training methodology and RD optimization for bit allocation. Firstly, we explain our network architectures. The training details are subsequently discussed, and finally the two pass encoding scheme is described to solve the aforementioned bit allocation optimization.

A. Network Architectures

1) MBFN: To achieve arbitrary positioned SAI generation, the neighbor SAIs are utilized as LF structural context. As shown in Fig. 4, suppose the central SAI is missing and needs to be generated. The MBFN takes the neighboring SAIS (top, bottom, left and right) as input. The fusion process contains four different branches with identical architecture. Each branch is composed of four convolution layers with 3×3 receptive field and 32 feature maps for each layer respectively. PReLU acts as the nonlinearity function for all convolution layers except for the last layer. Moveover, each branch utilizes the skip connection between input and output to take advantages of residual learning. The concatenation of all branches realizes the final fusion process, which generates the high order approximation (SAI_{approx}). Detailed configuration of MBFN is listed in Table II, where *Conv* denotes convolution layer, *Concate* is the the concatenation of each branch and the slop value (α_i in Eqn (5)) for PReLU is set to be 0.1. It is also worth mentioning that the Layer4 in Table II of Appendix is used for fusion.

2) RGN: After SAI_{approx} is obtained, the final estimation of the target SAI can be generated by RGN, which consists of six convolutional layers to build the mapping between high order approximation and target view ($f : SAI_{approx} \rightarrow SAI_{target}$). In analogous to MBFN, the detail parameter settings of RGN are shown in Table III.

3) Discriminator: To distinguish the proposed LF-GAN with previous CNN based methods, the adversarial training strategy is adopted since adversarial loss can naturally preserve the sharp edges as well as preventing the over-smoothing of detail texture. Therefore, the adversarial training can benefit the estimation and generation of slight disparity among neighbor SAIs while the conventional CNN based method only aims at reconstructing image in terms of L_2 loss. In our implementation, the discriminator contains six convolution layers followed by two dense connected layers (with dimensions 128 and 1 respectively). The details are provided in Table IV where *fc* denotes fully connected layer. It is worth mentioning that the batch normalization is configured for each convolution layer after activation to accelerate training.

B. Training Details

Since the proposed LF-GAN consists of three different components (MBFN, RGN, Discriminator) and is based on adversarial learning, to reduce the instability and optimize the performance, the progressive training method is adopted. Specifically, the component-wise training for each individual module is first performed, and the joint training with all components can be achieved. In particular, we first train the MBFN module with the Euclidean distance. Subsequently, the RGN and discriminator are trained with the loss defined in Eqn (7). Finally, we initialize each component with pre-trained weights to jointly optimize the whole system with Eqn (9).

1) Training Data: Let us denote each training sample as (ϕ, y_{GT}), where ϕ contains four blocks from four neighboring SAIs respectively, and y_{GT} is the corresponding ground truth block in the central SAI (as shown in Fig. 4). We train LF-GAN system to obtain the best estimation of y_{GT} by using $\phi(m-1,n), \phi(m+1,n), \phi(m,n-1)$ and $\phi(m,n+1)$ as inputs, where the $\phi(m-1,n), \phi(m+1,n), \phi(m,n-1)$ and $\phi(m,n+1)$ represent the blocks from neighboring SAIs to the top, bottom, left and right of $\phi(m,n)$ respectively. It should be noted that m and n are SAI index in LF-4D structure, where $m \in \{1,\ldots,x\}, n \in \{1,\ldots,y\}$ in Fig 1(a), and x, y are the angular resolution dimensions. The resolution of each SAI is 340×496 in our experiments. In the training phase, we divide SAIs into 36×36 sub-images with no overlap to train the whole network. During the testing phase, the entire SAIs are processed. It should be noted that we train different models for different QP values and all the context SAIs are compressed SAIs which can guarantee the consistency between encoder and decoder.

2) Boundary SAIs: It is worth noting that some of the context SAIs may be unavailable for the boundary SAIs in LF-4D structure (Fig. 8(a)), i.e., the SAIs between POC-0 and POC-1 in Fig. 6(a). For better illustration, we classify the boundary cases into six different categories, which are illustrated in Fig. 8(d). There could be only three neighboring SAIs available and one SAI unavailable (d.1-d.4 in Fig. 8(d)) or two SAIs available and two missing (d.5-d.6 in Fig. 8(d)). To elegantly address this issue, we utilize the linear prior of the existing three SAIs to approximate the unavailable SAI. Specifically, the available (blue rectangle in Fig. 8(d)) SAIs are averaged to obtain the unavailable SAIs (red rectangle in Fig. 8(d)) with following relationship,

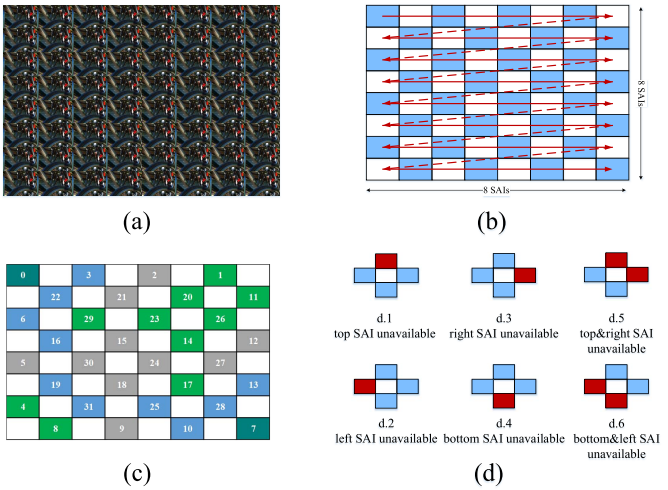


Fig. 8. (a) Illustrations of LF-4D structure, and the inner 8×8 LFs are used in our experiments; (b) sparsely sampled SAIs, and each small rectangle indicates one SAI in (a). Blue SAIs are needed to be compressed while white ones are generated at the decoder side by proposed LF-GAN using decoded neighboring SAIs; (c) hierarchical coding order of pseudo-sequence; (d) boundary case: only two or three SAIs are available (blue), and we use the average of the three blue decoded SAIs to fill the red rectangle.

$$SAI_{unavailable} = \frac{\sum_{i=1}^{h} SAI_{available}}{h}, \tag{21}$$

where h denotes the number of available SAIs (in our framework, $h = 3$ or 2 in $d.1 - d.4$ or $d.5 - d.6$ respectively). By using the linear prior, we can guarantee four input SAIs for LF-GAN.

3) Solver: Regarding the parameters of solvers, the default initializer in Keras [53] is used for the weights and bias initialization. All the training data are scaled between [0, 1] to avoid over-fitting and gradient explosion. To optimize the objective function Eqn (9), we utilize Adam [54] as the optimizer with learning rate 0.0001. Since the hyper-parameters are of vital influence on the convergence and performance of deep networks, we set the two hyper-params of Adam, β_1 and β_2 to be 0.9 and 0.999 respectively. Moveover, the hyper-parameters $\lambda_i, i = \{1, 2, 3\}$ in Eqn (9) are set to be 0.3, 0.65, 0.05 empirically, which can ensure the pixel-level reconstruction quality as well as the displacement between neighboring SAIs. Within our training, we utilized the 36×36 training patches and the batchsize for training is set to be 128. For each model, 60, 000 training iterations are needed for the LF-GAN to get its convergence. The whole learning framework is implemented based on the widely used DL library Tensorflow [55] and Keras [53] as back-end and front-end respectively.

C. Optimal Bit Allocation

To achieve high compression efficiency with optimal bit allocation, we utilize the two-pass encoding mechanism to obtain the model parameters in Eqn (20). For the first encoding pass, in which the sampled SAIs and unsampled SAIs are respectively encoded with 4 different QPs ($QP_{sp} = 18, 22, 28, 34$, and $QP_{usp} = 21, 27, 33$), generating 4 data points for R-Q model and D-Q model of sampled SAIs, and 12 points for unsampled SAIs. Therefore, the parameters $(a, b, c, \beta, \gamma, \tau, \mu, \theta, \phi, p_0, p_1, k_0, k_1)$ in Eqn (19) can be obtained using data regression for Eqn (13)–(16). Hence, the parameters for optimal bit allocation of sampled and unsampled SAIs can be determined. In the second coding pass, the two pseudo-sequences (sampled SAIs and unsampled SAIs) are encoded with the calculated q_{sp} and q_{usp}.

VI. Experimental Results

To validate the performance of the proposed GAN-based LF coding framework, the coding efficiency for LF images is tested and compared with the state-of-the-art learning based LF image compression method. In particular, we first analyze the RD performance and compare the LF SAI synthesis ability among multiple DL models for LF SAI interpolation, and then conduct the empirical analyses during training and present the complexity of the proposed LF-GAN.

A. Experiment Configurations

In our experiments, the EPFL LF image dataset [56] (Class A), Stanford light field dataset [57] (Class B) and HCI dataset [58] (Class C) are chosen for training and evaluation. Specifically, nine images from EPFL dataset, three images from Stanford dataset and four images from HCI dataset are selected for testing. It should be noted that there is no overlap between training set and test set. For the LF image pre-processing, the lenslet images in these datasets are firstly decomposed into LF-4D structure to obtain SAIs. To achieve fair comparison with [8] in terms of angular resolution of the LF images, only the internal 8×8 SAIs (depicted in Fig. 8(a)) are used for processing and compression because of the significant decomposition distortion of the boundary SAIs. For the sparse sampling, as shown in Fig. 8(b), each small rectangle represents one SAI. The SAIs marked with blue are those views to be sampled while the white ones are generated by LF-GAN at decoder side. Moreover, the $bitdepth$ for the LF pseudo-sequence is 8-bit and the color space format is YUV420.

For the proposed scheme, we firstly sparsely sample the SAIs in the LF-4D structure (as shown in Fig. 8) and compress those SAIs using the video codec HEVC (HM-14.0) with the proposed coding structure (Fig. 8(c)) and bit allocation scheme. Then, the residue SAIs are obtained and also compressed with the proposed coding structure and bit allocation method. Hence, the bitstream consists of the compressed sampled SAIs and the residue data. Subsequently, we generate the intermediate unsampled SAIs at decoder side by LF-GAN with the decoded SAIs as neighboring context. To achieve final reconstruction, the residue information is added into the generated SAIs.

B. Coding Efficiency Comparisons

The RD performance in terms of BD-PSNR and BD-rate [59] for each test LF image is shown. The PSNR value of luminance channel (PSNR_Y) is the averaged value for all SAIs according to [23],

$$PSNR_Y = \frac{1}{8 \times 8} \sum_{m=1}^{8} \sum_{n=1}^{8} PSNR[m][n], \tag{22}$$

The PSNR value of each SAI is calculated as follows,

$$PSNR[m][n] = 10 * \log_{10}[\frac{(1 << bitdepth - 1)^2}{MSE}],$$
$$MSE = \frac{1}{h \times w} \sum_{i}^{w} \sum_{j}^{h} (SAI_{rec}[i][j] - SAI_{ori}[i][j]))^2, \tag{23}$$

where h, w are spatial resolution of each SAI, and SAI_{rec} and SAI_{ori} denote the reconstructed SAI and uncompressed SAI respectively.

The coding performance BD-PSNR and BD-rate are listed in Table I. To generate the HEVC baseline (HM-14.0), we utilized two default coding structures low-delay P (LDP) and random access (RA) to compress both sampled SAIs and unsampled SAIs. From the last five columns of Table I, we can see that the proposed approach achieves significant coding gain with respect to HEVC. Multiple different anchors are compared in our experiments. For the LDP configuration, the proposed method obtains 32.6% BD-rate reduction on average and up to 40% bit-rates can be saved. While the performance for RA configuration, the proposed method achieves 14.8% BD-rate savings and 0.66dB BD-PSNR gain on average. To illustrate the effectiveness of proposed method, we also compare with the state-of-the-art learning based LF image compression methods. In general, the proposed LF-GAN based LF image coding framework outperforms the state-of-the-art learning based LF image compression approach [8] with overall 4.9% BD-rate reduction and 0.15dB BD-PSNR gain. We also compare the proposed approach with the pseudo-sequence based algorithm in [36]. And our method outperforms the algorithm in [36] with 8.1% BD-rate reduction over three different datasets. It is worth noting that the coding efficiency for three different datasets (Class A-C) remains consistent which show that the generalization ability of the LF-GAN is promising.

To better understand the performance of proposed approach, the RD curves of each test image are also provided in Fig. 9. Clearly, we can observe that LF image compression performance is significantly improved in a wide bit range by using the proposed GAN based view synthesis coding framework.

C. Quality Comparison of Different SAI Generation Algorithms

To better illustrate the advantage of the proposed method when generating the unsampled SAIs, we provide the subjective and objective ablation comparisons between the SAIs

TABLE I

RATE-DISTORTION PERFORMANCES IN TERMS OF BD-RATE (%) AND BD-PSNR (dB) OF PROPOSED GAN BASED LF IMAGE CODING FRAMEWORK

LF Images		Anchor: [8]		Anchor: [8] w/o RDO		Anchor: [36]		Anchor: HEVC LDP		Anchor: HEVC RA	
		BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR	BD-rate	BD-PSNR
Class A: EPFL Dataset	Ankylosaurus_Diplodocus_1	-1.5	0.05	-21.7	0.77	-14.9	0.72	-16.2	0.55	-6.8	0.20
	Bikes	-3.5	0.09	-21.6	0.69	-7.8	0.20	-42.0	9.00	-15.5	0.47
	Danger_de_mort	-6.3	0.22	-24.8	1.04	-10.8	0.28	-44.5	2.19	-28.4	1.24
	Desktop	-33.1	0.70	-22.3	0.94	-8.7	0.12	-26.9	0.58	10.1	-0.16
	Flowers	11.68	-0.48	-7.0	0.36	-11.9	0.54	-24.5	1.17	-1.3	0.11
	Fountain_Vincent_2	-5.6	0.16	-23.2	0.84	-9.0	0.31	-40.8	1.65	-20.4	0.70
	Friends_1	-0.6	0.00	-16.3	0.55	-19.0	0.55	-25.5	0.87	-0.3	0.03
	Stone_Pillars_Outside	-3.1	0.12	-29.0	1.05	-18.1	0.53	-42.2	1.64	-27.5	0.85
Class B: Stanford Dataset	Aloe	-10.0	0.37	-28.7	1.12	-9.1	0.31	-47.9	2.31	-31.7	1.38
	Vegetables	-4.6	0.16	-18.0	0.65	-10.2	0.22	-40.0	1.73	-11.1	0.40
	Orchid_Purple	-3.1	0.09	-14.8	0.47	-3.1	0.08	-37.6	1.41	-8.2	0.26
Class C: HCI Dataset	Bedroom	-1.7	0.07	-12.9	0.34	-3.6	0.08	-23.6	0.92	-18.4	1.04
	Bicycle	-5.6	0.29	-13.4	0.66	-5.9	0.18	-30.1	2.45	-13.2	1.09
	Herbs	-2.2	0.11	-5.2	0.54	-3.3	0.19	-21.5	1.35	-18.3	0.37
	Origami	-3.8	0.05	-12.3	0.45	-4.3	0.04	-18.0	0.55	-15.1	0.95
Class A		-5.6	0.11	-20.8	0.78	-12.5	0.41	-32.8	1.27	-11.3	0.43
Class B		-5.9	0.21	-20.5	0.75	-7.4	0.20	-41.7	1.82	-17.0	0.68
Class C		-3.3	0.13	-11.0	0.50	-4.3	0.12	-23.3	1.32	-16.2	0.86
Overall		-4.9	0.15	-17.4	0.68	-8.1	0.24	-32.6	1.47	-14.8	0.66

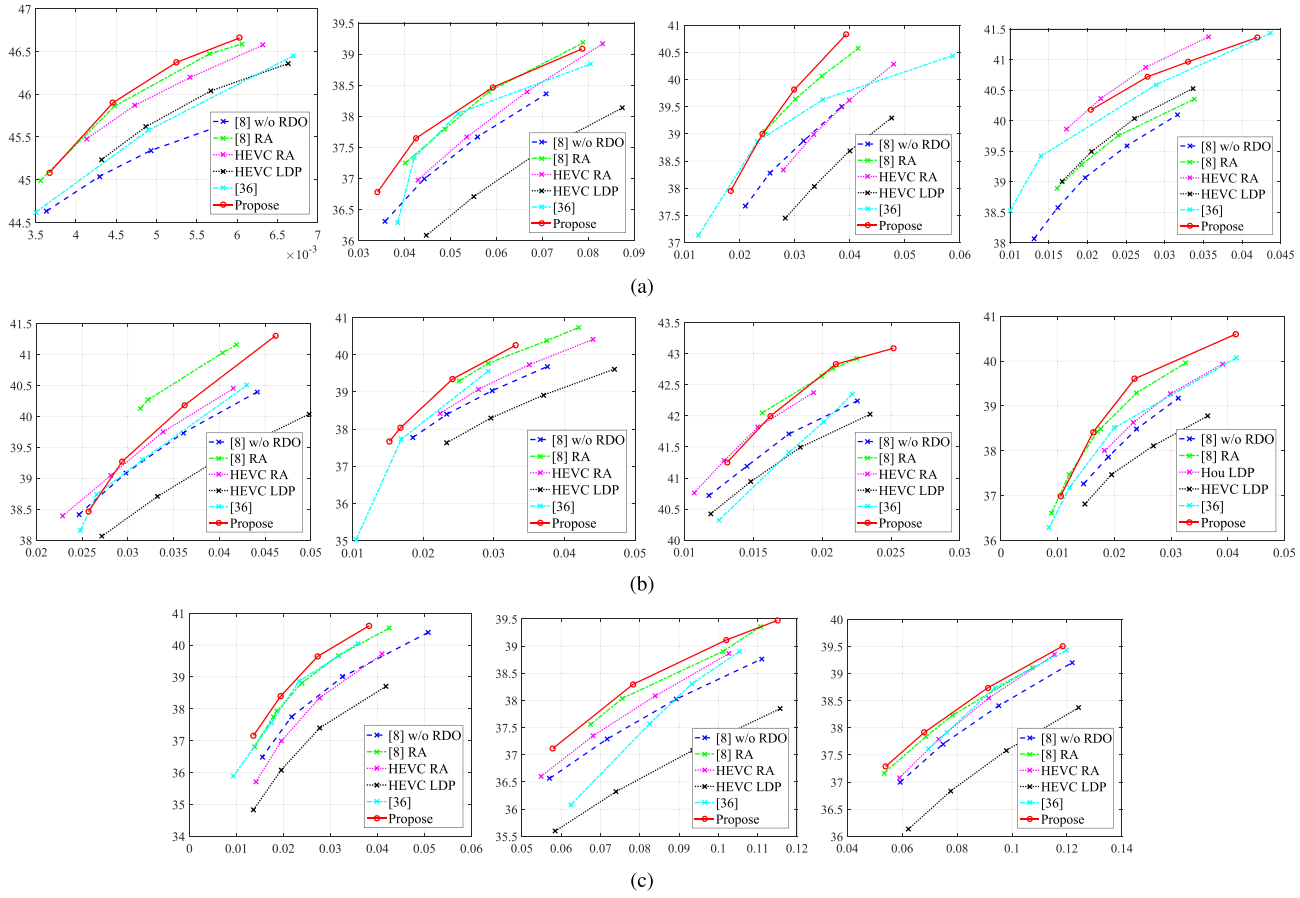


Fig. 9. The rate-distortion curves of each test image: (a) *Ankylosaurus_Diplodocus_1*, *Bikes*, *Danger_de_mort*, *Desktop*; (b) *Flowers*, *Fountain_Vincent_2*, *Friends_1*, *Stone_Pillars_Outside*; (c) *Aloe*, *Vegetables*, *Orchid_Purple*.

generated by different DL models in this section. Particularly, three algorithms are analyzed, the proposed method, the algorithms in [8] and [44]. We should note that we control all other variables to make the comparison fair and persuasive. Furthermore, in this ablation study, the test images are free of compression such that we could exclude the nuisance introduced by compression artifacts.

The central view of the 8×8 LF is utilized for illustration. We could easily observe from Fig. 10 that the proposed method achieves better perceptual quality with more the textural details when comparing with [7] and [44]. In particular, our models tend to keep structural consistency for the image content and with less over-smooth phenomena.

D. Analysis and Discussions

Regarding the computational complexity, we record the running time of GAN based view synthesis on a single PC, which is with Windows 10–64 bit system with Intel(R) Core(TM) i5 7300HQ and 8 GB memory and the GPU is NVIDIA GTX 1050Ti. The version of Tensorflow [55] and Keras [53] are 1.4.0 and 2.1.2 respectively. The average running time of all test LF images for GAN based view synthesis is

(a) (b) (c) (d)

Fig. 10. Visual quality comparison of the generated SAI for test LF image *Desktop*: (a) original; (b) by proposed (SSIM: 0.9921); (c) by [7] (SSIM: 0.9456); (d) by [44] (SSIM: 0.7123).

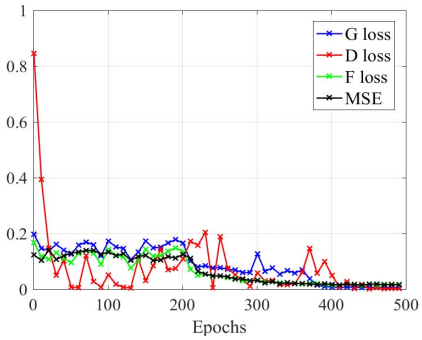


Fig. 11. The summary of different loss when training the adversarial network. G-loss: loss of RGN; D-loss: loss of Discriminator; F-loss: loss of MBFN; MSE: loss between label and the output of RGN.

52.6334 seconds. It is obvious that there is a clear margin for real-time LF image codec to use such framework, and the future work will focus on the inference optimization of the network by using model compression and acceleration.

Another interesting topic is the stability issue when training the RGN and discriminator of LF-GAN, which has been a widely discussed problem in deep learning community. We plot the loss values of different components in our LF-GAN in Fig. 11: G-loss (blue) is the loss of RGN, D-loss (red) denotes the loss of Discriminator, F-loss (green) is the loss of MBFN and MSE (black) denotes the loss between the output of RGN and the label. We can see that the G-loss, F-loss and MSE quickly gets their convergence after 200 epochs of training and the curves become flat. However, the D-loss does not show similar behaviors. Although finally the discriminator gets a relative good convergence, the unstable issue during training may affect the performance of LF-GAN. Therefore, a potential future research direction may focus on the stable training for GAN.

VII. Conclusion

The novelty of this paper lies in that we address the LF image compression problem based on the advanced GAN based view synthesis to improve the coding efficiency. In particular, the GAN based view synthesis network, LF-GAN, with an unsupervised visual perceptual learning model to generate the unsampled SAIs, is adopted. By analogy with adversarial learning, the proposed LF-GAN, which is composed of a CNN based architecture with input fusion network, generator and discriminator, can reliably generate the contents of an arbitrary positioned SAI conditioned on its surroundings. Since the SAIs based pseudo-sequence has different RD characteristics with natural videos, we propose the typical hierarchical coding structure for pseudo-sequence based coding to facilitate the generation of intermediate SAIs in LF structure. For better reconstruction for SAIs, the residue data of generated unsampled SAIs are compressed and transmitted to the decoder-side. To further enhance the coding efficiency, the joint optimal bit allocation scheme is also proposed for sampled SAIs and unsampled ones. We quantitatively demonstrate the effectiveness of our proposed LF image compression framework with GAN based view synthesis. Extensive experimental results show that the proposed method outperforms the state-of-the-art learning based coding approach.

Appendix
Detail Parameters of LF-GAN

The network parameter settings for MBFN, RGN and Discriminator are provided in Tables II, III and IV, respectively.

TABLE II
PARAMETER SETTINGS FOR EACH BRANCH OF THE PROPOSED MBFN

Index	Layer 1	Layer2	Layer3	Layer4
Layer Type	Conv	Conv	Conv	Conv
Receptive Field	3×3	3×3	3×3	3×3
Pading	1	1	1	1
Feature Map Number	32	32	32	32
Activation (α=0.2)	PReLU	PReLU	PReLU	PReLU

TABLE III
PARAMETER SETTINGS OF THE PROPOSED RGN

Index	Layer1∼Layer5	Layer6
Layer Type	Conv	Conv
Receptive Field	3×3	3×3
Pading	1	1
Feature Map Number	32	1
BatchNorm	Yes	No
Activation	PReLU (α=0.1)	-

TABLE IV
PARAMETER SETTINGS OF THE PROPOSED DISCRIMINATOR

Index	Layer1∼Layer6	Layer7	Layer8
Layer Type	Conv	fc	fc
Receptive Field	3×3	-	-
Pading	1	-	-
Feature Map Number	32	128-dim	1-dim
BatchNorm	Yes	-	-
Activation	PReLU (α=0.1)	PReLU (α=0.1)	Sigmoid

Acknowledgment

The authors would like to thank the guest editor and anonymous reviewers for their constructive comments that significantly helped us in improving the presentation of the manuscript of this paper and also thank the authors of [8], [36] for kindly sharing their implementation for performance comparison.

REFERENCES

[1] (2016). *Lytro*. [Online]. Available: https://www.lytro.com/

[2] (2016). *Raytrix*. [Online]. Available: https://www.raytrix.de/

[3] G. Wu *et al.*, "Light field image processing: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 926–954, Oct. 2017.

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

[5] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. 23rd Annu. Conf. Comput. Graph. Interact. Techn.*, 1996, pp. 31–42.

[6] T.-C. Wang, J.-Y. Zhu, N. K. Kalantari, A. Efros, and R. Ramamoorthi, "Light field video capture using a learning-based hybrid imaging system," *ACM Trans. Graph. (TOG)*, vol. 36, no. 4, p. 133, 2017.

[7] N. K. Kalantari, T.-C. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph.*, vol. 35, no. 6, p. 193, 2016.

[8] J. Hou, J. Chen, and L.-P. Chau, "Light field image compression based on bi-level view compensation with rate-distortion optimization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 2, pp. 517–530, Feb. 2019.

[9] M. S. K. Gul and B. K. Gunturk, "Spatial and angular resolution enhancement of light fields using convolutional neural networks," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2146–2159, May 2018.

[10] H. Zheng, M. Guo, H. Wang, Y. Liu, and L. Fang, "Combining exemplar-based approach and learning-based approach for light field super-resolution using a hybrid imaging system," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 2017, pp. 2481–2486.

[11] S. Heber, W. Yu, and T. Pock, "Neural EPI-volume networks for shape from light field," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2271–2279.

[12] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image-Based Rendering*. New York, NY, USA: Springer, 2008.

[13] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 6, pp. 974–988, Jun. 2009.

[14] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 606–619, Mar. 2014.

[15] H.-G. Jeon *et al.*, "Accurate depth map estimation from a lenslet light field camera," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1547–1555.

[16] Z. Lin and H.-Y. Shum, "A geometric analysis of light field rendering," *Int. J. Comput. Vis.*, vol. 58, no. 2, pp. 121–138, 2004.

[17] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.* Reading, MA, USA: Addison-Wesley, 2000, pp. 307–318.

[18] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3746–3754.

[19] 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.

[20] L. Shi, H. Hassanieh, A. Davis, D. Katabi, and F. Durand, "Light field reconstruction using sparsity in the continuous Fourier domain," *ACM Trans. Graph.*, vol. 34, no. 1, p. 12, 2014.

[21] (2016). *Icme 2016 Grand Challenge: Light-Field Image Compression*. [Online]. Available: https://mmspg.epfl.ch/ICME2016GrandChallenge_1/

[22] (2017). *ICIP 2017 Grand Challenge I: Light Field Image Coding*. [Online]. Available: http://2017.ieeeicip.org/GC1.asp/

[23] (2017). *JPEG Pleno Final Call for Proposals on Light Field Coding*. [Online]. Available: https://jpeg.org/items/20170208_cfp_pleno.html/

[24] C. Conti, P. Nunes, and L. D. Soares, "Hevc-based light field image coding with bi-predicted self-similarity compensation," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.

[25] R. J. S. Monteiro, P. J. L. Nunes, N. M. M. Rodrigues, and S. M. M. Faria, "Light field image coding using high-order intra-block prediction," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1120–1131, Oct. 2017.

[26] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Coding of focused plenoptic contents by displacement intra prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 7, pp. 1308–1319, Jul. 2016.

[27] R. Zhong, S. Wang, B. Cornelis, Y. Zheng, J. Yuan, and A. Munteanu, "L1-optimized linear prediction for light field image compression," in *Proc. Picture Coding Symp. (PCS)*, 2016, pp. 1–5.

[28] B. Girod, "Efficiency analysis of multihypothesis motion-compensated prediction for video coding," *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 173–183, Feb. 2000.

[29] M. Rerabek, T. Bruylants, T. Ebrahimi, F. Pereira, and P. Schelkens, "CFP of the ICME light field compression challenge," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, 2016, pp. 1–4.

[30] I. Viola, M. Řerábek, 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.

[31] X. Jin, H. Han, and Q. Dai, "Image reshaping for efficient compression of plenoptic content," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1173–1186, Oct. 2017.

[32] X. Jin, H. Han, and Q. Dai, "Plenoptic image coding using macropixel-based intra prediction," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3954–3968, Aug. 2018.

[33] M. Magnor and B. Girod, "Data compression for light-field rendering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 338–343, Apr. 2000.

[34] C.-L. Chang, X. Zhu, P. Ramanathan, and B. Girod, "Light field compression using disparity-compensated lifting and shape adaptation," *IEEE Trans. Image Process.*, vol. 15, no. 4, pp. 793–806, Apr. 2006.

[35] C. Jia *et al.*, "Optimized inter-view prediction based light field image compression with adaptive reconstruction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4572–4576.

[36] D. Liu, L. Wang, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2016, pp. 1–4.

[37] L. Li, Z. Li, B. Li, D. Liu, and H. Li, "Pseudo-sequence-based 2-D hierarchical coding structure for light-field image compression," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1107–1119, Oct. 2017.

[38] 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.

[39] Y. Li, M. Sjöström, R. Olsson, and U. Jennehag, "Scalable coding of plenoptic images by using a sparse set and disparities," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 80–91, Jan. 2016.

[40] S. Zhao, Z. Chen, K. Yang, and H. Huang, "Light field image coding with hybrid scan order," in *Proc. Vis. Commun. Image Process. (VCIP)*, 2016, pp. 1–4.

[41] F. Dai, J. Zhang, Y. Ma, and Y. Zhang, "Lenselet image compression scheme based on subaperture images streaming," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 4733–4737.

[42] C. Jia, Y. Yang, X. Zhang, S. Wang, S. Wang, and S. Ma, "Light field image compression with sub-apertures reordering and adaptive reconstruction," in *Proc. Pacific Rim Conf. Multimedia*. Harbin, China: Springer, 2017, pp. 47–55.

[43] Z. Zhao, S. Wang, C. Jia, X. Zhang, S. Ma, and J. Yang, "Light field image compression based on deep learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[44] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. S. Kweon, "Learning a deep convolutional network for light-field image super-resolution," in *Proc. IEEE Int. Conf. Comput. Vis. Workshop*, Dec. 2015, pp. 24–32.

[45] R. Garg, B. G. V. Kumar, G. Carneiro, and I. Reid, "Unsupervised CNN for single view depth estimation: Geometry to the rescue," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands: Springer, 2016, pp. 740–756.

[46] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.

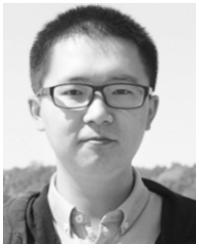
[48] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[49] J. Johnson, A. Gupta, and L. Fei-Fei, "Image generation from scene graphs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1219–1228.

[50] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. (2017). "Image-to-image translation with conditional adversarial networks." [Online]. Available: https://arxiv.org/abs/1611.07004

[51] K. Gregor, F. Besse, D. J. Rezende, I. Danihelka, and D. Wierstra, "Towards conceptual compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3549–3557.

[52] S. Wang, S. Ma, S. Wang, D. Zhao, and W. Gao, "Rate-GOP based rate control for high efficiency video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1101–1111, Dec. 2013.
[53] F. Chollet, "Keras," Tech. Rep., [Online]. Available: URL: https://keras.io
[54] D. P. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980
[55] M. Abadi *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[56] M. Rerabek and T. Ebrahimi, "New light field image dataset," in *Proc. 8th Int. Conf. Qual. Multimedia Exper. (QoMEX)*, 2016, pp. 1–2.
[57] B. Wilburn *et al.*, "High performance imaging using large camera arrays," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 765–776, Jul. 2005.
[58] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *Proc. Asian Conf. Comput. Vis.* Taipei, Taiwan: Springer, 2016, pp. 19–34.
[59] G. Bjontegaard, *Calculation of Average PSNR Differences Between RD-Curves* document VCEG-M33, 2001.



Chuanmin Jia (S'18) received the B.Sc. degree in computer science and technology from the Beijing University of Posts and Telecommunications, China, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science, Peking University, Beijing, China. He was a Visiting Student with the Video Lab, New York University, New York, NY, USA, in 2018. His research interests include video coding, machine learning, and light field image compression.

He received the Best Paper Award from the Pacific-Rim Conference on Multimedia in 2017 and the *IEEE Multimedia Magazine* in 2018, respectively.

Xinfeng Zhang (M'16) received the B.S. degree in computer science from the Hebei University of Technology, Tianjin, China, in 2007, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2014. From 2014 to 2017, he was a Research Fellow with the Rapid-Rich Object Search Lab, Nanyang Technological University, Singapore. He is currently a Post-Doctoral Fellow with the School of Electrical Engineering System, University of Southern California, Los Angeles, CA, USA. His research interests include image and video processing and image and video compression.

Shanshe Wang received the B.S. degree from the Department of Mathematics, Heilongjiang University, Harbin, China, in 2004, the M.S. degree in computer software and theory from Northeast Petroleum University, Daqing, China, in 2010, and the Ph.D. degree in computer science from the Harbin Institute of Technology. He held a post-doctoral position at Peking University, Beijing, from 2016 to 2018. He joined the School of Electronics Engineering and Computer Science, Institute of Digital Media, Peking University, where he is currently a Research Assistant Professor. His current research interests include video compression and image and video quality assessment.

Shiqi Wang (M'15) received the B.S. degree in computer science from the Harbin Institute of Technology in 2008 and the Ph.D. degree in computer application technology from Peking University in 2014. From 2014 to 2016, he was a Post-Doctoral Fellow with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. From 2016 to 2017, he was a Research Fellow with the Rapid-Rich Object Search Laboratory, Nanyang Technological University, Singapore. He is currently an Assistant Professor with the Department of Computer Science, City University of Hong Kong. He has authored over 40 technical proposals to ISO/MPEG, ITU-T, and AVS standards. His research interests include image/video compression and analysis and quality assessment.

Siwei Ma (M'03–SM'12) received the B.S. degree from Shandong Normal University, Jinan, China, in 1999, and the Ph.D. degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005. He held a post-doctoral position at the University of Southern California, Los Angeles, CA, USA, from 2005 to 2007. He joined the School of Electronics Engineering and Computer Science, Institute of Digital Media, Peking University, Beijing, where he is currently a Professor. He has authored over 200 technical articles in refereed journals and proceedings in image and video coding, video processing, video streaming, and transmission. He is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *Journal of Visual Communication and Image Representation*.