LF2MV: Learning An Editable Meta-View Towards Light Field Representation

Menghan Xia, Jose Echevarria, Minshan Xie, and Tien-Tsin Wong

Abstract—Light fields are 4D scene representations that are typically structured as arrays of views or several directional samples per pixel in a single view. However, this highly correlated structure is not very efficient to transmit and manipulate, especially for editing. To tackle this issue, we propose a novel representation learning framework that can encode the light field into a single meta-view that is both compact and editable. Specifically, the meta-view composes of three visual channels and a complementary meta channel that is embedded with geometric and residual appearance information. The visual channels can be edited using existing 2D image editing tools, before reconstructing the whole edited light field. To facilitate edit propagation against occlusion, we design a special editing-aware decoding network that consistently propagates the visual edits to the whole light field upon reconstruction. Extensive experiments show that our proposed method achieves competitive representation accuracy and meanwhile enables consistent edit propagation.

Index Terms—Light field, Compact representation, Editing propagation, Representation learning.

1 Introduction

4D Light fields model incoming light rays hitting the camera sensor at different locations and from different angles, which allows them to capture complex geometries and material appearances faithfully [1], [2]. This rich representation enables applications in synthesis of novel viewpoints and refocusing [3], or geometry [4] and material analysis [5]. Given raw light fields are typically structured as an array of multiple images from different viewpoints, or multiple directional samples per pixel, such memory-intensive collections of rays are challenging to handle or edit efficiently. Existing light field compression [6], [7] and editing techniques [8], [9], [10] were proposed separately. However, although the latter enables new kinds of edits only possible with light fields, the amount of highly correlated data needed in run-time makes naive approaches inefficient. Moreover, the available tools are too disruptive for typical edits with respect to the well established 2D image editing workflows. So, a more memory-efficient light field representation that is easier to digest by existing tools would be desirable.

To tackle this problem, we propose to represent the light field as a single meta-view, which consists of visual channels that capture the visual content of the scene (i.e. the central RGB view) and a complementary meta channel that stores other view-dependent information like geometric or appearance properties. The meta channel allows the original light field to be reconstructed with high fidelity. Apparently, the meta-view representation is compact thanks to the efficiently encoded meta channel. Furthermore, we require the meta channel to be compatible to edits on the visual channels, such that the editing effects on the visual channels can be consistently propagated to the other views when the full light field is reconstructed. In practice, this two features make our representation suitable for lightweight desktop or mobile editing interfaces, which requires memory no more than a typical 2D RGBA image and supports popular 2D image processing algorithms by nature (Fig. 1). We believe it makes an important step to promote light fields consumption at scale, as currently prevented by large bandwidth and specialized complex editing tools.

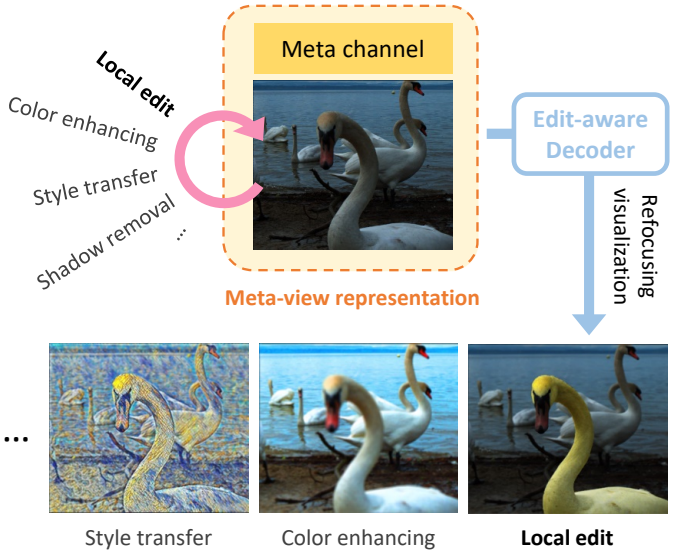


Fig. 1: Application illustration. The learned meta-view representation enables light filed edits to be as simple as 2D image manipulation, which is memory-efficient and supports most image processing algorithms (e.g. color transfer, style transfer, shadow removal, etc). Thanks to its compactness, the process can be performed on lightweight mobile devices or via online cloud computation.

To achieve those features, we propose a representation learning framework to learn an edit-aware encoding-decoding scheme (see Fig. 2). Specifically, the encoding sub-

- *Menghan Xia, Minshan Xie and Tien-Tsin Wong are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, HK. {mhxia, msxie, ttwong}@cse.cuhk.edu.hk.*
- *Jose Echevarria is with the Adobe System Inc, US. echevarr@adobe.com.*

network converts the light field into a single meta channel, which together with the RGB central view, composes the meta-view representation and allows the decoding subnetwork to recover the full light field. The two subnetworks are then jointly trained to ensure the learned representation to be restorable to the original light field. However, it is non-trivial to reconstruct the desired light field from a possibly edited representation. To enable the edit-aware reconstruction, we propose to decompose the decoding process into three components: (i) feature separation that extracts individual view information from the meta channel; (ii) disparity recovery that aims to warp the visual channels to other views; (iii) fusion synthesis that restores the view-dependent visual content from the corresponding feature map and then synthesizes the target view based on the warped result. Note that, the fusion synthesis component not only utilizes the information encoded from the input, but also predicts the potential editing effects for occlusions based on the surrounding context. We train our model using publicly available light field datasets under self-supervision. Particularly, we introduce several edits to the visual channels during training, so as to force the model to handle editing propagation.

To evaluate our method, we collected a group of light fields from publicly available datasets, which were captured with Lytro cameras under various settings. Extensive quantitative and qualitative evaluations are conducted. Results show that our method achieves competitive reconstruction accuracy along with compactness and consistent editing propagation effects. In summary, our main contributions include:

- We present a novel light field representation learning framework, which enables the light fields to be represented with a compact and editable meta-view. It allows low-bandwidth transmitting and efficient visual edits using standard 2D image editing tools, with the complex geometric properties preserved.
- We propose an effective edit-aware decoding network, which facilitates accurate reconstruction as well as consistent edit propagation.
- Our method opens a promising direction for consumer-level light field processing, which may promote light field applications at scale.

2 Related Works

2.1 Light Field Reconstruction

Previous works on light field reconstruction focus on synthesizing a dense light field from a sparser set of samples. Traditional methods mainly formulate the view synthesis as an optimization problem to generate the novel views directly [11], [12] or estimate disparities to obtain them through warping [13], [14]. These methods tend to present ghosting and tearing artifacts when the input views are sparse. Recently, deep learning techniques have been explored for synthesis of dense views [15], [16]. Kalantari et al. [17] synthesize a novel view with two sequential CNNs that estimate disparity for warping and refine the colors jointly. Wu et al. [18] focus on recovering the high frequency details of linearly upsampled epipolar-plane images (EPIs), where a blur-deblur scheme is employed to tackle the information asymmetry problem. To deeply characterize spatial-angular clues, Yeung et. al. [19] employ spatial-angular alternating convolutions within a residual learning framework. Different from these methods requiring supervision of dense light fields, Ni et al. [20] utilize inter-view cycle consistency to enable the unsupervised learning of a light field from two input views, where occlusions are compensated through a forward-backward warping scheme. Reducing the amount of input views to one, Srinivasan et al. [21] explore the task of synthesizing a light field from a single image, which can obtain impressive results for specific scenes. In all of them, the ill-posed sparse-to-dense reconstruction struggles to recover or generate information invisible in the input views due to occlusions.

Another direction for light field reconstruction deals with single or multiple coded images that are generated at data capturing stage and targets for accurate reconstruction from them [22], [23], [24]. Anyhow, such coded images are usually generated via hand-crafted procedures that are usually limited in expressiveness and accuracy, and moreover the data-embedded views are infeasible to be edited afterwards. More recently, multiplane images (MPIs) have been proposed as an effective representation for light fields [25], [26], [27], with very efficient novel view synthesis for scenes with complex geometries and materials. However, it is not clear how to edit such overabundance of layers that, although effective for parallax effects, do not model the implicit geometries and materials faithfully.

2.2 Light Field Editing

Most previous works focus on individual editing tasks such as retargeting [28], morphing [29] or completion [30]. These specialized editing methods are difficult to generalize and unify in a single framework, so users cannot perform multiple typical edits with the same tool. A different line of research aims for more general edits instead. Seitz et al. [31] propagate local edits across multiple views through a voxel-based light field representation. Jarabo et al. [32] propose a novel downsampling-upsampling propagation scheme to propagate sparse edits to the full light field based on an affinity function. Chen et al. [9] extend the versatile patch-based editing framework to the domain of light fields and enable several interesting new editing operations. Zhang et al. [10] decompose the central view into a set of layers at different depths, so existing patch-based edits can be applied to them. Recently, Beigpour et al. [33] present an intrinsic decomposition framework for editing the appearance of surfaces through various band shift operators. Aiming to exploit the 4D information in a light field, Jarabo et al. [8] propose and study novel interfaces and workflows. Different from all the methods above, we aim to obtain a both compact and editable meta-view through a representation learning framework, which allows light fields to be edited efficiently using existing 2D image editing tools.

2.3 Light Field Compression

There are typically two kinds of light field compression frameworks. The first line of technique aims to compress the

raw light field sensor data at acquisition. Such lenslet based intra-coding algorithms mainly work on self-similarity compensation prediction and local linear embedding [34], [35], [36]. Some methods employ sensor-adaptive transformation and reshaping for better compression efficiency [37]. Recently, deep learning models are utilized to design the lenslet encoding pattern that benefits the compressive sensing reconstruction importantly [24]. The other category of technique considers the 4D light field representation for compression. Considering the highly correlated views of light fields, Conti et al. [38] first introduce the concept of self-similarity for compensation prediction, which is integrated into H.264/AVC [39] standard to efficiently compress light field images. Besides, Pseudo-Temporal Sequence (PTS) based methods are proposed to organize the light field images into sequence and compress them with video coding standards to reduce data redundancy [40], [41]. There are multiple scanning orders proposed for PTS generation, and the coding structure and bit-rate allocation are optimized for better rate-distortion performance [42], [43].

3 Meta-View Representation Learning

Given a 4D light field, we aim to encode it as a meta-view that consists of a single meta channel and several visual channels (like an RGB image), through a representation learning framework. The four-channel meta-view is more compact than the original 4D representation. Importantly, the edits on the visual channels are required to propagate across the full views consistently when the light field is reconstructed.

3.1 Problem Formulation

We denote a structured 4D light field consisting of an array of $M \times N$ RGB views as $\mathbf{L} = \{\mathbf{I}_i\}_{i=1}^{M\times N}$, where $\mathbf{L}(\mathbf{u}, \mathbf{x})$ samples the light field at angular coordinate $\mathbf{u} = (u, v)$ and spatial coordinate $\mathbf{x} = (x, y)$, so $\mathbf{I}_i = \mathbf{L}(\mathbf{u}_i)$. As illustrated in Fig. 2, our framework consists of an encoding subnetwork (encoder) $\mathcal{E}$ and a decoding subnetwork (decoder) $\mathcal{D}$. The encoder takes a light field $\mathbf{L}$ as input, and generates a meta channel $\mathbf{Z}$ which is a float-valued 2D array of the same resolution as $\mathbf{I}_i$. We construct a tuple $\{\mathbf{Z}, \mathbf{I}_c\}$ as the intermediate representation, where $\mathbf{I}_c$ denotes the editable central view of $\mathbf{L}$. Inversely, the decoder $\mathcal{D}$ reconstructs a light field $\tilde{\mathbf{L}} = \{\tilde{\mathbf{I}}_i\}_{i=1}^{M\times N}$ from the edited representation $\{\mathbf{Z}, \tilde{\mathbf{I}}_c\}$, where $\tilde{\mathbf{I}}_c$ is the edited version of $\mathbf{I}_c$. Formally, they can be described as:

$$\mathbf{Z} = \mathcal{E}(\mathbf{L}) \quad (1)$$

$$\tilde{\mathbf{L}} = \mathcal{D}(\mathbf{Z}, \tilde{\mathbf{I}}_c). \quad (2)$$

Here $\mathcal{E}$ and $\mathcal{D}$ roughly work as a pair of inverse functions with their own trainable parameters. According to our goal, the reconstructed $\tilde{\mathbf{L}}$ should be as similar as possible to the target light field $\hat{\mathbf{L}}$, whose appearance is coherent with $\tilde{\mathbf{I}}_c$ while maintains the same scene geometry of $\mathbf{L}$. This process is challenging because of occlusions, since the meta channel is encoded from the original light field and agnostic to the potential edits on the visual channels. As a trivial case, $\tilde{\mathbf{I}}_c = \mathbf{I}_c$ implies no editing involved, so we have $\hat{\mathbf{L}} = \mathbf{L}$, which enables self-supervised learning. In Section 3.3, we will discuss how to train our model with edits propagation considered.

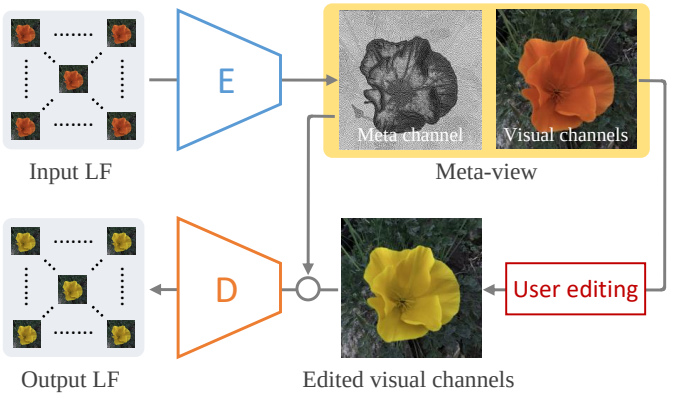
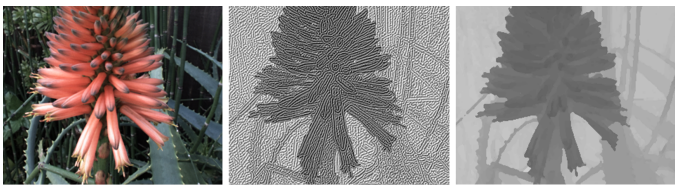


Fig. 2: System overview. Given an input light field, the encoder converts it into a single meta channel, which together with the visual channels (the central RGB view) serves as a compact and editable representation of the original light field, called *meta-view*. Users can edit the appearance of the visual channels using standard 2D editing tools. Then the edited visual channels and original meta channel are taken by the decoder to reconstruct the full light field with all the edits consistently propagated.

3.2 Representation Space

Our model learns to represent a light field as a meta-view consisting of the central view and a single meta channel. The central view covers the visual content of the light field mostly, and the encoder only needs to learn the complementary residual data, i.e. the meta channel, from the input light field. Therefore, the meta-view is embedded with the full information of the original light field, which is realized by the representation learning framework in Fig. 2. More specifically, the meta channel plays the role of recording the original light field information beyond the central view image, including: the disparities between the central view and other views, the invisible content from the central view, and other view-dependent information like non-Lambertian reflection. As illustrated in Fig. 3, the encoded meta channel roughly resembles the depth map of the central view but presents additional texture. This implies that apart from the depth/disparity information,



(a) Central view (b) Meta channel (c) Depth map of (a)

Fig. 3: Representation study. The central view (a) is used as the visual channels. The meta channel (b) roughly resembles the depth value of the central view but shows some extra structured patterns. The depth map (c) is estimated from the whole light field using [44].

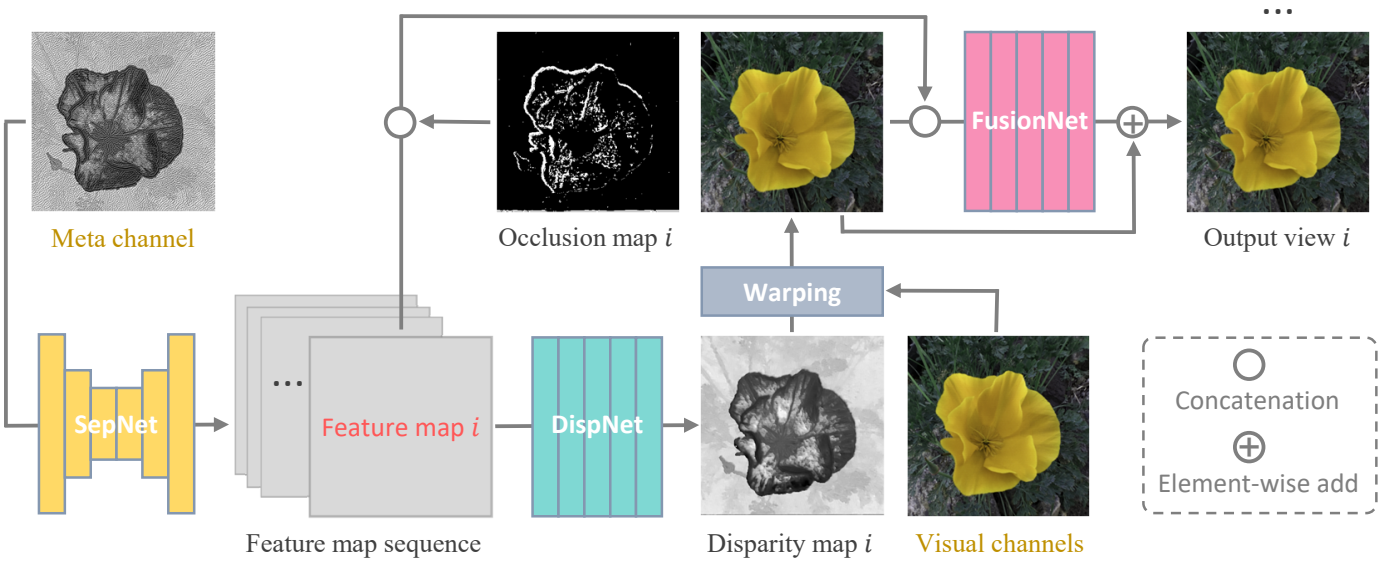


Fig. 4: Schematic diagram of the decoder. The input meta channel is decomposed into sequential features $\{\mathbf{F}_i\}_{i=1}^{M\times N}$ by *SepNet*. *DispNet* takes $\mathbf{F}_i$ as input to generate the corresponding disparity map $\mathbf{D}(\mathbf{u}_i)$. A warped view $\bar{\mathbf{I}}_i$ is generated by warping the visual channels with $\mathbf{D}(\mathbf{u}_i)$. Then, an occlusion mask is generated by checking inter-view disparity consistency, which along with $\bar{\mathbf{I}}_i$ and $\mathbf{F}_i$ are fed to *FusionNet* to reconstruct the targeted view i.

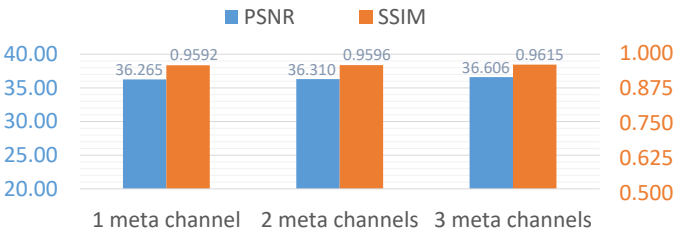


Fig. 5: Reconstruction accuracy against the number of meta channel. The numerical statistics are from the evaluation on the testing dataset (described in Section 4).

other information is implicitly encoded in the form of structured patterns. To verify this, we conduct an ablation study in Section 5.3, where the meta channel is replaced with the depth map of the central view and the reconstruction turns problematic.

Naturally, as the scale of input light field (e.g. spatial resolution or angular resolution) increases, the single meta channel may not have enough capacity to represent all the necessary information. In that case, we can increase the number of meta channels, for example, two or more meta channels will be generated. Alternatively, the visual channels could be used for data embedding, i.e. encoding the whole light field into the visual channels $\mathbf{I}^z = \mathcal{E}(\mathbf{L})$ via the encoder, which means no meta channel is used. This is feasible to serves as a more compact representation (as evaluated in Fig. 9). However, the encoded visual channels tolerates no edits, because the implicitly encoded information may get damaged. So, separation between visual and meta channels is preferred in our targeted scenario. In addition, for the case of 7×7 views in our dataset (detailed in Section 4), one meta channel achieves the best balance between efficiency and compactness, as justified by Fig. 5.

3.3 Editing-Aware Reconstruction

A U-shaped network is employed as the encoder, which learns a residual feature representation of the input. However, the decoder involves a more complex process that a light field is reconstructed from an intermediate representation undergone potential edits. The naive solution of modeling the decoder with a single network, suffers from poor generalization to the edits on the visual channels (see Fig. 14). Instead, we propose to decompose the decoder into several components: *SepNet*, *DispNet* and *FusionNet*, each of which learns a specific task. The schematic diagram is illustrated in Fig. 4. The architectures of the encoder and decoder are detailed in the supplementary material.

Feature Separation - *SepNet*. As stated in Section 3.2, the meta channel encodes all the information of the input light field, except for those in the visual channels. In order to reconstruct the light field views, we instruct the decoder to interpret that information as two categories: the inter-view disparity maps and other view-dependent information (e.g. occluded contents and non-Lambertian effects). Specifically, *SepNet* is utilized to decompose the meta channel $\mathbf{Z}$ into a sequence of feature maps $\{\mathbf{F}_i\}_{i=1}^{M\times N}$, where $\mathbf{F}_i$ is contains the exclusive information of view i. Then, each view $\tilde{\mathbf{I}}_i$ can be reconstructed from the feature map $\mathbf{F}_i$ and visual channels $\tilde{\mathbf{I}}_c$ independently.

Disparity Recovery - *DispNet*. We extract from $\mathbf{F}_i$ the disparity map $\mathbf{D}(\mathbf{u}_i)$ of view i through *DispNet*. $\mathbf{D}(\mathbf{u}_i, \mathbf{x})$ is a 2D vector, sliced from the 4D disparity field of $\mathbf{L}$, which denotes the disparity of pixel $\mathbf{L}(\mathbf{u}_i, \mathbf{x})$ with respect to its horizontal and vertical neighboring views. Specifically, given view $\mathbf{L}(\mathbf{u}_j)$ and the disparity map $\mathbf{D}(\mathbf{u}_i)$, we can obtain the warped view i via:

$$\bar{\mathbf{L}}(\mathbf{u}_i, \mathbf{x}) = \mathbf{L}(\mathbf{u}_j, \mathbf{x} + (\mathbf{u}_j - \mathbf{u}_i) \cdot \mathbf{D}(\mathbf{u}_i, \mathbf{x})). \quad (3)$$

With the disparity map $\mathbf{D}(\mathbf{u}_i)$, we can obtain the corresponding warped view by warping the visual channels

(a) Original visual channels (b) Edited visual channels (c) Inset of (b) (d) Recovered corner view

Fig. 6: Occlusion recovery with editing effect propagated. Compared to the edited visual channels (c), the patch of the reconstructed corner view (d) reveals the occluded content with the editing effect reasonably propagated.

$\mathbf{I}_c$. Obviously, the warped view $\bar{\mathbf{L}}(\mathbf{u}_i)$ will not be exactly the same as our target view $\hat{\mathbf{L}}(\mathbf{u}_i)$ because of occlusions and non-Lambertian effects. Following the idea of Ruder et al. [45], we estimate the occlusion regions by performing a forward-backward consistency check of the disparity. In particular, when warping $\mathbf{I}_c$ to view i, the occlusion map i can be approximated as:

$$\mathbf{O}(\mathbf{u}_i, \mathbf{x}) = ||\mathbf{D}(\mathbf{u}_i, \mathbf{x}) - \mathbf{D}(\mathbf{u}_c, \mathbf{x} + (\mathbf{u}_c - \mathbf{u}_i) \cdot \mathbf{D}(\mathbf{u}_i, \mathbf{x}))||_1, \tag{4}$$

where the greater the value of $\mathbf{O}(\mathbf{u}_i, \mathbf{x})$ is, the more likely the sample $\mathbf{L}(\mathbf{u}_i, \mathbf{x})$ is occluded in $\bar{\mathbf{I}}_c$, thus indicating a stronger necessity to be corrected by *FusionNet*.

Context Based Synthesis - *FusionNet*. To restore the occluded details and non-Lambertian effects, we further refine the warped view $\bar{\mathbf{I}}_i$ by *FusionNet* through residual learning. Particularly, $\mathbf{F}_i$ provides the necessary view-dependent information of the original light field , while the occlusion map $\mathbf{O}_i$ indicates the less reliable regions of the warped view $\bar{\mathbf{I}}_i$. Ideally, we would like to reconstruct the target view in a deterministic way. However, in the case of edited visual channels, it is substantially ambiguous how the edits should be propagated to those occluded regions of the target view. To infer the principle of edit propagation, the model should make good use of the context around the occlusions, and we propose to learn such knowledge from large-scale training pairs $\{\tilde{\mathbf{I}}_c, \hat{\mathbf{L}}\}$. Unfortunately, it is impractical to manually prepare a diversely edited version of the light field dataset, because the inter-view consistency is tricky to maintain. Instead, we propose to make use of some global editing operators $\mathcal{G}(\bullet)$, such as changing the hue, saturation, exposure, and contrast, whose parameters are sampled from an uniform distribution. The nice property of those operators is that the light field views can be processed individually and the inter-view consistency are preserved automatically. So, we collect the training pairs $\{\mathcal{G}(\mathbf{I}_c), \mathcal{G}(\mathbf{L})\}$ easily. In addition, although only naive global color manipulation is involved during training, our model generalizes well to allowing many popular image edits to the visual channels, as described in Section 5.2. The possible explanation is that those global edits and arbitrary local edits share the same essence of changing pixel values, and the model learns the propagation of edits across views. Fig. 6 show that our model propagates the edited colors to those occluded areas effectively.

3.4 Loss Function

The encoding and decoding subnetworks are jointly trained by minimizing a loss function that consists of warping consistency loss $\mathcal{L}_W$, disparity regularity loss $\mathcal{L}_D$, and reconstruction loss $\mathcal{L}_R$:

$$\mathcal{L} = \omega_1 \mathcal{L}_W + \omega_2 \mathcal{L}_D + \omega_2 \mathcal{L}_R, \tag{5}$$

where $\omega_1, \omega_2, \omega_3$ are the weighting coefficients of different loss terms. First, the warping consistency loss measures the pixel-wise difference between the light field warped from the central view $\mathbf{I}_c$ and the input light field as:

$$\mathcal{L}_W = \mathbb{E}_{\mathbf{L}_i \in \mathcal{S}}\{||\bar{\mathbf{L}}_i - \mathbf{L}_i||_1\}, \tag{6}$$

which actually poses constrains on the disparity maps generated from *DispNet*. The rationale lies in the fact that the baselines of light field cameras is very small, so the same object almost have the same appearance across different views. $||\bullet||_1$ means L1 norm and $\mathbb{E}_{\mathbf{L}_i \in \mathcal{S}}$ denotes the average operator over all the light fields in the training dataset $\mathcal{S}$.

Only with $\mathcal{L}_W$, the accuracy of disparity maps are not well guaranteed, because the textureless regions are insensitive to incorrect disparity under $\mathcal{L}_W$. It could be argued that errors in such regions are negligible if we only care about the correctness of the warped view. However, inaccurate disparity maps directly decrease the reliability of occlusion maps, thus complicates the training of *FusionNet* which takes as input the occlusion map as correction hints. So, we explicitly strengthen the disparity consistency between neighbor views with the regularity loss:

$$\mathcal{L}_D = \mathbb{E}_{\mathbf{L}_i \in \mathcal{S}}\{||\mathbf{D}(\mathbf{u}_i, \mathbf{x}) - \mathbf{D}(\mathbf{u}_i - \mathbf{1}, \mathbf{x} + \mathbf{D}(\mathbf{u}_i, \mathbf{x}))||_1\}, \tag{7}$$

which encourages the predicted disparity to be consistent across views and the estimated occlusions to be sparse. For efficiency, the disparity consistency of each view is only checked with three neighbor views, i.e. $\mathbf{1} = \{(0,1), (1,0), (1,1)\}$. Under this constraints, the light field views are related as a connected graph, where each view is affected by the rest indirectly.

Finally, we regulate the reconstructed light field $\tilde{\mathbf{L}}$ to be the same as the target one $\hat{\mathbf{L}}$ via the reconstruction loss:

$$\mathcal{L}_R = \mathbb{E}_{\mathbf{L}_i \in \mathcal{S}}\{\alpha||\tilde{\mathbf{L}}_i - \hat{\mathbf{L}}_i||_1 + \beta||\text{SSIM}(\tilde{\mathbf{L}}_i, \hat{\mathbf{L}}_i)||_1\}, \tag{8}$$

where $\text{SSIM}(\bullet, \bullet)$ measures the average SSIM [46] over all the views of the light field. $\alpha = 1.0$ and $\beta = 0.02$ are used to balance the magnitudes of the two loss terms. We find that SSIM helps reconstruct accurate details, especially in the disocclusion regions.

As introduced above, the three terms, i.e. $\mathcal{L}_W$, $\mathcal{L}_D$, and $\mathcal{L}_R$, play different roles in training supervision. We empirically set $\omega_1 = 0.5, \omega_2 = 0.01$ just to balance the numerical magnitudes of $\mathcal{L}_W$ and $\mathcal{L}_D$. In addition, to make the model focus more on the final reconstruction accuracy, we recommend $\omega_3 > \omega_1$ and adopt $\omega_3 = 1.0$ in our experiments while other values may also work.

4 Implementation Details

Dataset preparation. Our dataset is built from two publicly available datasets: the Stanford Lytro Lightfield Archive [47] and the dataset from Kalantari et al. [17],

both of which were captured with Lytro Illum cameras. By removing those repeated or low-quality samples manually, 304 light fields from [47] and 102 light fields from [17] are used in our dataset. Each light field has 376×541 spatial samples and 14×14 angular samples, but only the central 7×7 grid of angular samples are used in our experiments because many peripheral angular samples are outside the camera aperture. Among the 406 light fields, 326 are randomly selected as the training dataset and the remaining 80 are used as the testing dataset. In the training dataset, each light field sample, containing 49 RGB views, is randomly cropped into 20 patches of 128×128 pixels. Also, we augment each patch by flipping it horizontally or rotating it $180°$, where the image order are adjusted accordingly to preserve the epipolar geometry of the light field. Finally, with textureless patches filtered out by an automatic procedure that computes average gradient, we get $14,447$ light field samples to train our model.

Training scheme. As stated in Section 3.4, both the encoding subnetwork and decoding subnetwork are trained jointly. However, we find that directly training all the networks from scratch suffers from low-speed convergence. This might be explained by the fact that at the beginning, the disparity maps from *DispNet* are mostly random noise, which thus causes problematic occlusion maps, confusing the training of *FusionNet*. To avoid this situation, we propose to train the network in two stages. In the first stage, all the networks excluding *FusionNet*, are trained jointly for the first $10,000$ iterations, so as to warm up the *DispNet*. In the second stage, all the networks including *FusionNet* are trained for another $50,000$ iterations. Specifically, we optimize our model using the Adam optimizer [48], with the learning rate fixed as 0.0002 in the first stage and linearly decreased to its 1.0% in the second stage.

We implement our model using Pytorch[1], a Python-based open-source deep learning framework. All the experiments are ran on a server equipped with two NVIDIA Geforce GTX 1080 Ti. The training consumes roughly 48 hours. A light field with 7×7 angular samples for 376×541 spatial samples takes about 0.043s (seconds) to encode the meta channel, and 2.056s to reconstruct a full light field (0.042s per view) from it. Our source code is publicly available at: https://github.com/MenghanXia/LF2MV.

5 Experimental Results

The propose method is evaluated in three aspects: reconstruction accuracy along with compactness, editing propagation consistency, and ablation study on key designs. Additional results and video visualizations are available in the supplementary material.

5.1 Representation Accuracy

Comparison with Baselines. We evaluate the representation accuracy of our meta-view with the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM). As no prior arts solving our problem, we choose to compare with existing techniques that reconstruct light fields from visual representation, despite they were devised for different purposes. First, we compare with Inagaki's method [24] that represents a light field as one or a few coded images by learning micro-lens weighting masks. We adopt the pre-trained model provided by the authors, which reconstructs a light field of 5×5 views from two coded images. Since our model is trained for light fields of 7×7 views, we simply take the central 5×5 views of the reconstructed light field for comparison. The results are presented in Table 1, which shows our method outperforms Inagaki's method at a large margin. The main reason is that Inagaki's method regulates the encoding scheme as polynomial combination of the input light field views, which severely limits the solution space. Instead, our encoding scheme is adaptively learned under the guidance of reconstruction accuracy. To visualize the reconstructed light field, we present a corner view along with epipolar slices in Fig. 7.

TABLE 1: Reconstruction accuracy over our testing dataset (central 5×5 views used). Higher PSNR/SSIM is better.

Method	PSNR		SSIM	
	Mean	Stddev	Mean	Stddev
Inagaki et al. [24]	32.676	0.8508	0.9211	0.0095
Ours	**37.232**	2.0698	**0.9638**	0.0083

(a) A corner view of the ground truth LF (b) Inagaki et al. [24] (c) Ours

Fig. 7: Visual comparison with Inagaki et al. [24]. The insets show the reconstruction details while the EPIs evaluate both the inter-view consistency and parallax correctness. The boundary colors of insets denote different methods (grey denotes the ground truth), and yellow arrows indicate problematic regions.

Srinivasan et al. [21] synthesize a light field from an ordinary 2D image, which serves as a naive baseline of ours. Since synthesizing a light field from a single image is extremely ill-posed, the model trained on specific type of scene generalizes poorly to new types of scenes. For fair comparison, we evaluate on the dataset used in their paper, which consists of 100 light fields of flowers. The quantitative results are tabulated in Table 2, and a typical example for visual comparison is illustrated in Fig. 8. As expected, Srinivasan's method [21] can not recover accurate details that are occluded in the input image, which are filled with dark pixels or surrounding textures instead.

Compactness Evaluation. As our method represents a light field as a meta-view, we evaluate the representation compactness with respect to the original view-array representation. Note that, our method does not target for

1. Pytorch: https://pytorch.org/

TABLE 2: Reconstruction accuracy over Flowers dataset [21] (7×7 views). Higher PSNR/SSIM is better.

Method	PSNR		SSIM	
	Mean	Stddev	Mean	Stddev
Srinivasan et al. [21]	33.365	3.2325	0.9115	0.0380
Ours	**40.944**	1.8097	**0.9719**	0.0062

(a) A corner view of the ground truth LF (b) Srinivasan et al. [21] (c) Ours

Fig. 8: Visual comparison with Srinivasan et al. [21]. The insets show the reconstruction details while the EPIs evaluate both the inter-view consistency and parallax correctness. The boundary colors of insets denote different methods (grey denotes the ground truth), and white arrows indicate problematic regions.

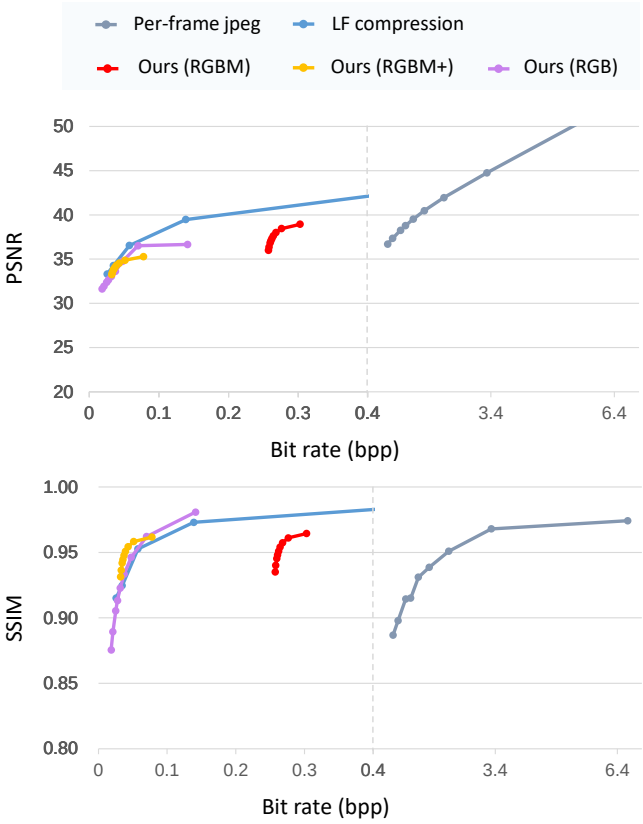


Fig. 9: Rate-distortion evaluation on different light field representations. Two metrics are used to measure the light field reconstruction accuracy: PSNR against bit rate (top) and SSIM against bit rate (bottom). For better view, the horizontal axis is divided at bpp=0.4 and visualized with different scalars.

light field compression but it is orthogonal to compression algorithms that can be applied to our meta-view for further storage efficiency. For comparison, we evaluate on the Rate-Distortion performance (R-D curve) of different representations, including (i) view-array; (ii) our meta-view, both of which are applied with compression algorithms. In particular, the view-array representation is compressed with two kinds of compression algorithms, i.e. per-view JPEG and a state-of-the-art light field compression algorithm [49] respectively. As for our meta-view, we apply JPEG to the visual channels and adopt lossless Huffman codes [50] onto the meta channel. To show the compression potential of our representation, we extend our method to another two variants of higher compactness. *meta-view (RGBM+)*: consists of visual channels and a meta channel that is encoded as 8-bit bitmap and tolerates JPEG compression. *meta-view (RGB)*: consists of the learned visual channels only that has the meta information embedded, which also tolerates JPEG compression. The JPEG-robustness embedding implementation is quite straightforward, i.e. introducing a JPEG noise layer (as [51] did in their paper) during training. The details are presented in the supplementary material.

Fig. 9 compares the R-D curves, plotted with fidelity measurement PSNR and SSIM against bit number per-pixel (bpp). Following the practice of [49], both the PSNR and SSIM are computed in YCrCb color space and the weighted sum of three channels are adopted (the weight coefficients are 6:1:1). Note that, all of our variants are compressed by applying JPEG on the visual channels, and particularly for ours and *ours (RGBM+)*, the meta channel is compressed by Huffman codes and JPEG respectively. The JPEG compression on images are conducted by traversing the quality factor set $\{50, 60, 70, 75, 80, 85, 90, 95, 100\}$. We find that our meta-view (red curve) has a notably better compactness than the common view-array representation (gray curve). Even when the special light field compression algorithm [49] adopted (blue curve), the more compact variants of our method (purple and yellow curves) still achieve very competitive performance. Anyway, although ours (RGB) has the best compression performance, it can not support edits because the vulnerable meta information is embedded among the pixel values. So, there is a trade-off between compactness and editability when selecting a variant of our representations for applications. Nevertheless, for a specific scene, our method never offer a nice way to tune the balance between them because it is a either-or choice.

5.2 Visual Edit Propagation

Our meta-view allows edits on the visual channels, just like editing an ordinary RGB image. Anyhow, since the meta channel is not editable, our method requires the edited appearance still matching the original scene geometry. Although this leaves out basic edits like scaling, copying or inpainting, we argue that our method already supports plenty of advanced and useful edits, even including some complex edits not possible with previous light field editing

Fig. 10: Showcase of light field reconstruction from edited meta-views. The edited visual channels and one of the reconstructed corner views are provided to check the edit propagation. The attached horizontal and vertical EPIs (for the scanlines marked in white) visualize the inter-view consistency and parallax correctness.

tools. For instance, the supported visual edits include, global/local color manipulation (e.g. exposure, contrast, saturation, hue, etc), local texture modification, and most image processing algorithms (e.g. color transfer, denoising, dehazing, shadow removal, style transfer, etc).

Qualitative evaluation. We illustrate some typical examples in Fig. 10, where the light fields are reconstructed from the meta-views whose visual channels are edited through: (a) global and local color adjustment with *Photoshop*, (b) shadow removal algorithm [52], (c) photorealistic color transfer [53], (d) low-lightness enhancemen [54], (e) photo cartoonization [55], and (f) style transfer technique [56] respectively. As there is no ground truth for comparison, we evaluate the propagated edits by visually inspecting the inter-view consistency and parallax correctness. For each example, we illustrate one of the challenging corner views that have the largest disparities with respect to the central view, and the horizontal and vertical epipolar plane images (EPIs) of spatial segments, with comparison to the original light field. By comparing the EPIs, we observe that the reconstructed light fields have very similar geometric structures to the original light fields, since the parallaxes are reflected by the slopes of the color strips. Besides, the insets of the corner views illustrate the reconstruction quality in local regions, where no artifacts or visual inconsistencies are observed. Anyhow, the toon style transfer simplifies structural details while the artistic style transfer tends to introduce extra textures to the results, and thus they may cause some reasonable inconsistency between the EPIs of the edited light fields and the original ones. Readers are recommended to watch the supplementary video for better inspection. More results are available in the supplementary material.

Quantitative evaluation. Due to the absence of ground truth, we adopt two indirect measurements to perform evaluation over the testing dataset. Firstly, we make use of some global color operations, i.e. changing the hue, saturation, exposure and contrast, to construct paired data for evaluation. Specifically, we randomly apply one of these operations (sampling the parameters from a uniform distribution) to the views of each light field individually, which can approximately serve as the ground-truth edited version. Then, we evaluate the reconstruction accuracy of the meta-view that consists of the edited central view and the meta channel encoded from the original (non-edited) light field. The average PSNR and SSIM are **36.099** and

TABLE 3: Edit propagation accuracy on typical examples shown in Fig. 10. The control group with no edits applied is provided for reference. Higher PSNR is better.

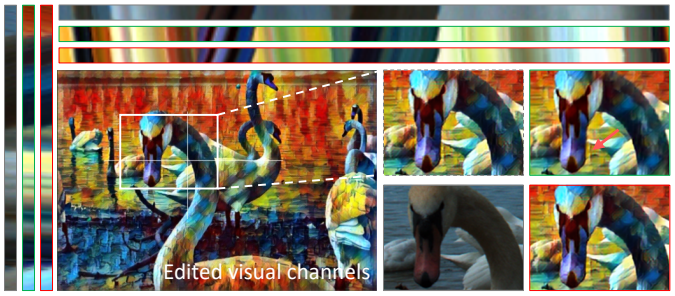
Example	With Edit		No Edit	
	PSNR	SSIM	PSNR	SSIM
Color edit	33.068	0.9733	33.144	0.9736
Shadow removal	34.159	0.9748	34.226	0.9752
Color transfer	35.509	0.9726	35.538	0.9726
Lightness enhance	35.342	0.9600	35.504	0.9619
Toon style transfer	41.092	0.9873	41.185	0.9861
Artistic style transfer	39.865	0.9755	40.109	0.9761
Average	**37.663**	**0.9739**	**37.790**	**0.9742**

$\mathbf{0.9539}$ respectively, only slightly lower than the reconstruction accuracy without edits involved (i.e. $\mathbf{36.265}$ and $\mathbf{0.9592}$). Anyhow, the global color edit is just one category of the edits supported by our method. To evaluate over more diverse edits, e.g. those demonstrated in Fig. 10, we propose to utilize an indirect measurement: given a light field $\mathbf{L}$ and its edited visual channels $\tilde{\mathbf{I}}_c$, we run the encoding and decoding recurrently for twice: $\mathbf{L}^{rec} = \mathcal{D}(\mathcal{E}(\tilde{\mathbf{L}}), \mathbf{I}_c)$ with $\tilde{\mathbf{L}} = \mathcal{D}(\mathcal{E}(\mathbf{L}), \tilde{\mathbf{I}}_c)$. Then, we can measure the difference between the recurrently reconstructed light field $\mathbf{L}^{rec}$ and the original light field $\mathbf{L}$ conveniently. The results are tabulated in Table 3, with a control group that has $\tilde{\mathbf{I}}_c = \mathbf{I}_c$ provided for reference. It shows that our edit reconstruction almost has the same accuracy as the reconstruction without any edit involved, which suggests the decent edit propagation performance in some sense.

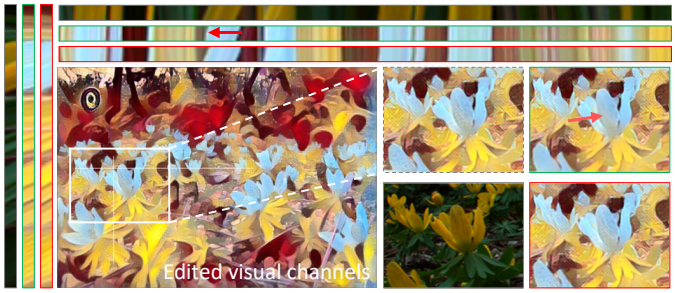
Comparison with disparity based edit propagation. Recently, the light field style transfer (LFST) [57] is proposed to utilize disparity-based warping for inter-view consistence guidance. Note that, LFST is optimized over each light field separately, so it is not only edit-specific but inefficient in comparison with our method. Fig. 11 illustrates two visual examples, from which we can find LFST is inferior to our method in three aspects: (i) the edit in occluded regions is less accurately propagated, as evidenced by the corner view in (a); (ii) the artistic texture is smoothed by propagation, as evidenced by the corner view in (b); (iii) the disparity of the resultant LF is inaccurately preserved, as evidenced by the EPI in (b). For quantitative evaluation, we follow the practice of LFST [57] by estimating the depth map from the resultant light field and measuring the deviation w.r.t that from the input light field. We evaluate on those samples used in [57], where four artistic styles are adopted. The results are compared in Table 4, showing the consistent superiority of our method in edit propagation accuracy. The qualitative results are presented in the supplementary material.

TABLE 4: Quantitative evaluation on the depths estimated from stylized light fields (four styles are used). Lower MAE means better accuracy.

Method	MAE of estimated depth			
	Candy	*Mosaic*	*Rain-princess*	*Udnie*
LFST [57]	17.327	17.703	16.732	16.939
Ours	**8.587**	**8.014**	**9.201**	**8.493**



(a) Occlusion reconstruction



(b) Texture propagation

Fig. 11: Edit propagation comparison between LFST [57] and ours. The insets with solid boundaries show the reconstructed corner views and EPIs of different methods: **grey** denotes the ground truth without edits, **green** denotes LFST, and **red** denotes ours. The red arrows indicate problematic regions.

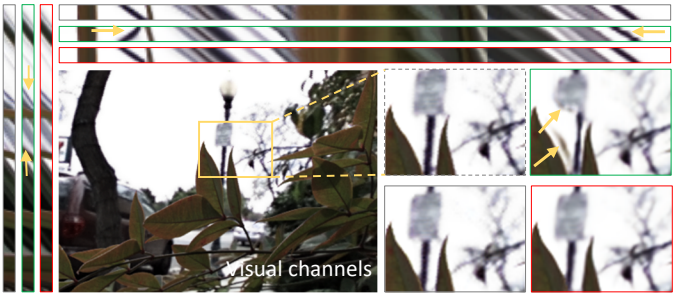
5.3 Ablation Study

Meta Channel. We have the meta channel record the necessary information that enables the original light filed to be reconstructed from the meta-view. However, one may argue that the meta channel might be no more than a depth map and all the other missing information could be predicted from data prior. To study this issue, we construct a baseline by replacing the learned meta channel with a central-view depth map, namely the RGB-D representation, for comparative analysis. Particularly, the depth map is estimated from the whole light field through the state-of-the-art method [44]. For fair comparison, we utilize a CNN with the same architecture as our decoder to reconstruct the light field from the RGB-D representation under the same supervision defined in Eq. 5. The quantitative results are tabulated in Table 5, showing that the representation accuracy of RGB-D is inferior to ours. However, the gap is not very large since the inter-view occlusion of Lytro captured light fields (narrow baseline) is relatively mild. Still, the qualitative comparison illustrated in Fig. 12 reveals the major drawbacks of the RGB-D baseline with respect to our proposed meta-view.

Firstly, the reconstruction accuracy of RGB-D highly relies on the quality of the depth map, however depth estimation from light fields is challenging itself, especially in the cases of busy background and fine structures. As a result, distortion and ghost effect are induced in the reconstructed views, as shown in Fig. 12(a). Besides, a depth map is unable to record the accurate depth of object boundaries because pixels of object boundary tend to be a

TABLE 5: Quantitative results of ablation study. The reconstruction accuracy is compared over the testing dataset, while the editing propagation is compared over the edit cases shown in Fig. 10. Higher PSNR/SSIM is better.

Method	Reconstruction		Edit Propagation	
	PSNR	SSIM	PSNR	SSIM
RGB-D baseline	34.139	0.9454	34.541	0.9657
Direct decoding	36.505	0.9620	36.143	0.9723
No warping	**37.771**	**0.9740**	37.086	0.9731
No separation	34.574	0.9551	34.080	0.9579
Ours	36.265	0.9592	**37.663**	**0.9739**



(a) Thin structure

(b) Occlusion

Fig. 12: Reconstruction comparison between RGB-D and our representation. The insets with solid boundaries show the reconstructed corner views and EPIs of different methods: **grey** denotes the ground truth, **green** denotes RGB-D baseline, and **red** denotes ours. The yellow arrows indicate problematic regions.

mixture of foreground and background and are associated with multiple depths. Therefore, it makes no sense to take a depth map explicitly for information representation. In contrast, our meta channel is adaptively learned through a representation learning framework. It is flexible to encode any necessary information implicitly in order to reconstruct the original light field, including occlusions, which in contrast is impossible for a depth map. In Fig. 12(b), the occluded baluster is recovered from our representation but fails to be reconstructed by the RGB-D baseline. Naturally, these limitations of RGB-D baseline retain in the editing propagation results, as demonstrated in Fig. 13.

Decoding Network. Our proposed decoding network consists of three special components that are employed to perform edit-aware reconstruction. The key designs include per-view feature separation and intermediate warping operation. To study their necessity, we construct three baselines for comparison, including (i) *Direct decoding*: we adopt a

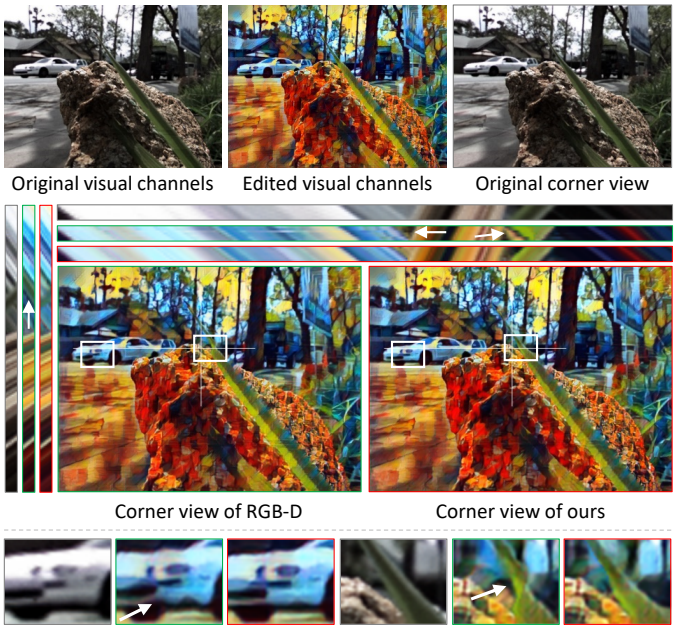
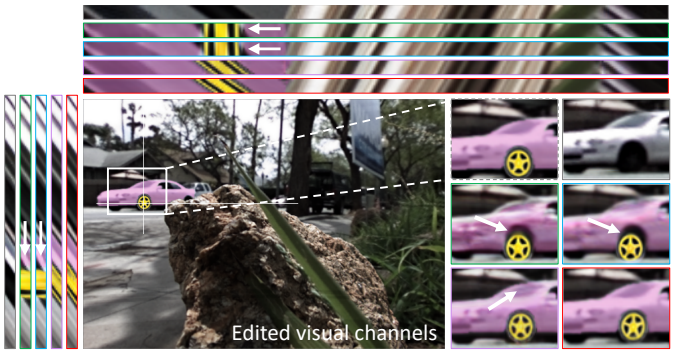
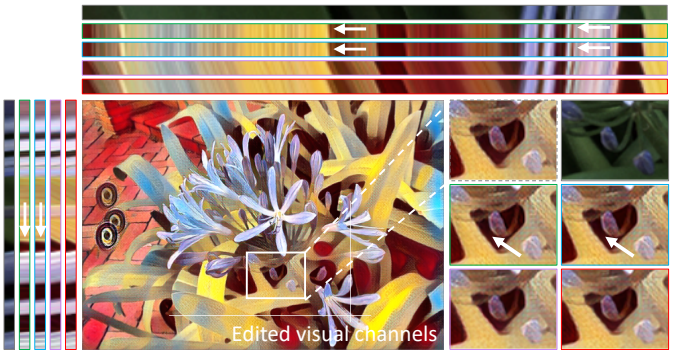


Fig. 13: Edit propagation comparison between RGB-D and our representation. The insets with color boundaries show the reconstructed corner views and EPIs of different methods: *grey* denotes the ground truth without edits, *green* denotes RGB-D baseline, and *red* denotes ours. The white arrows indicate problematic regions.

single network (the same as the encoder) to reconstruct the target light field from the meta-view representation directly; (ii) *No separation*: *SepNet* and *DispNet* are combined together to generate disparity maps of all views and then refines each warped view through *FusionNet*; (iii) *No warping*: we modify *DispNet* to generate per-view features instead of the disparity map, which together with the visual channels are fed to *FusionNet*. The quantitative results are tabulated in Table 5. We find that the intermediate warping operation harms the reconstruction accuracy, as *Direct decoding* and *No warping* have higher accuracy than *No separation* and ours. This might be explained by the more flexible decoding mechanism of the task-adaptive fitting than that of the hand-crafted warping. On the contrary, the intermediate warping helps the model generalize on diverse edits, especially for those significantly changed colors or textures. It suggests that instructing the model to work per prior knowledge benefits in generalization. Fig. 14 illustrates two typical examples. For those baselines without warping adopted, texture pasting causes noticeable visual artifacts and parallax failure (a). Severe texture modification, e.g. by artistic style transfer, almost removes the parallax between the reconstructed views, as evidenced by the EPIs (b). In addition, we further evaluate the edit propagation performance quantitatively (the examples in Fig. 10 are used) as tabulated in Table 5. We observe that the design of per-view feature separation helps improve reconstruction accuracy no matter whether edits involve or not. In summary, our proposed decoding network demonstrates comprehensive advantages over all the baselines for edit-aware reconstruction.



(a) Texture pasting



(b) Style transfer

Fig. 14: Edit propagation comparison between different decoding networks. The insets with solid boundaries show the reconstructed corner views and EPIs of different methods: **grey** denotes the ground truth without edits, **blue** denotes *direct decoding*, **green** denotes *no warping*, **purple** denotes *no separation*, and **red** denotes ours. The white arrows indicate problematic regions.

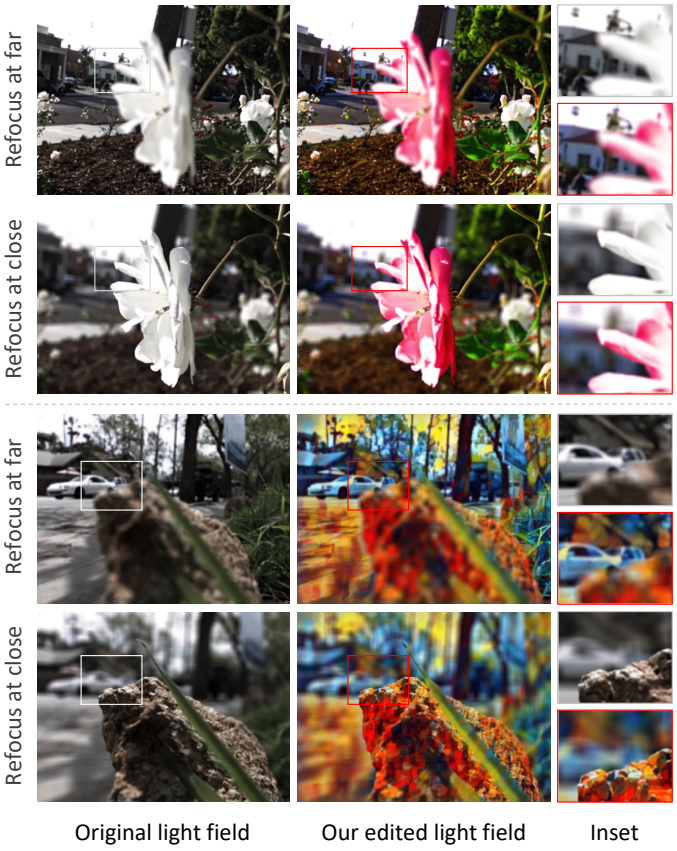


Fig. 15: Refocusing application of our reconstructed light fields. For each example, the results of focusing at close and far distance are illustrated. The insets offer clearer inspection.

5.4 Application Study

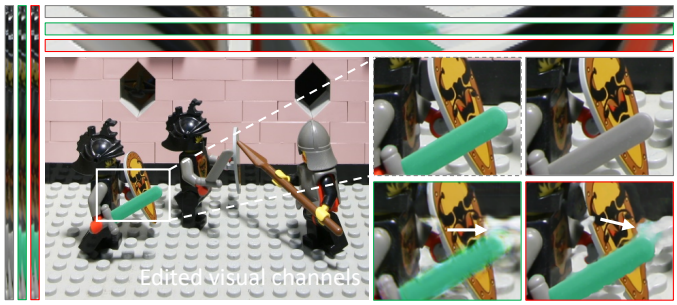
Computational Refocusing. A typical application of light fields is computational refocusing. As our method enables lots of fancy 2D image processing algorithms applicable to light fields, it is interesting to utilize those edited light fields for refocusing application. Fig. 15 demonstrates two examples, each of which presents both close and far focusing results. Comparing to the original light fields, our reconstructed light fields support similar refocusing performance. In practice, our proposed technique may even benefit to artistic creation. As it is not easy to manually simulate realistic refocusing effects, our technique enables physically correct refocusing effects on cartoon photos or artistic photos.

Light Filed with Wide Baseline. So far, we have demonstrated the effectiveness on the light fields captured by Lytro cameras. It remains to be studied the applicability to wide-baseline light fields, i.e. captured by camera gantry. Considering the publicly available gantry dataset is far from sufficient for training a CNN model, we instead use the synthetic dataset *POV-Ray LF dataset* [58] for applicability study. *POV-Ray LF dataset* contains light fields that associates higher resolution (11×11 views and 640×480 per view) and larger baseline (the max disparity reaching to 20 pixels). In total, 500 light fields are collected, of which 400 samples are randomly selected for training and the rest 100 samples are used for evaluation. Due to the GPU memory limitation, we train and evaluate the models on light fields with 7×7 and 9×9 views respectively. Particularly, two variants of decoding network are used in our model, i.e. the direct decoding baseline and our proposed edit-aware decoding network. Table 6 shows the results. We observe that the reconstruction accuracy of large-baseline light fields decreases notably with respect to that of those Lytro captured ones. It is because the occlusion regions are much larger and hence challenges the data embedding and recovery. Fig. 16 illustrates two examples, including a synthetic sample and a real-world scene captured by camera gantry. We observe that both the direct decoding baseline and our edit-aware decoding network cause visual artifacts, especially in those occlusion regions. But the visualization of EPIs evidences that our method can still recover the disparities accurately. In summary, our model fails to reproduce the good performance on large-baseline light fields as on small-baseline ones, but we expect more advanced network blocks, such as deformable convolution [59] and Transformer [60], to bridge the gap by promoting the disparity-aware embedding and decoding effectiveness. As large-baseline light fields are mainly used for depth estimation while small-baseline light fields are better for refocusing, our current method still has promising applications to the popular Lytro camera data.

TABLE 6: Reconstruction accuracy on larger-baseline light fields. Higher PSNR/SSIM is better.

Method	7×7		9×9	
	PSNR	SSIM	PSNR	SSIM
Direct decoding	28.829	0.8742	28.158	0.8569
Ours	28.768	0.8900	28.043	0.8734

(a) Synthetic data



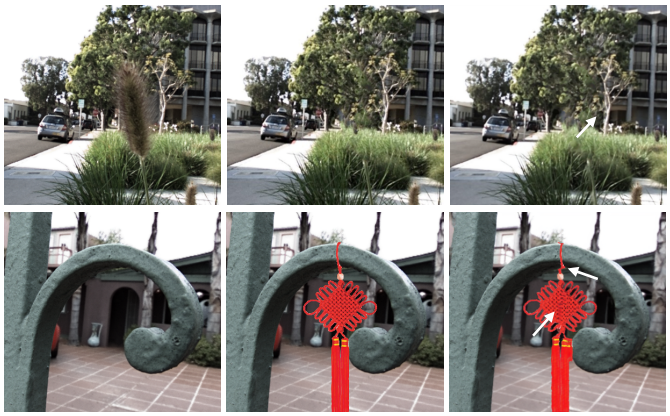
(b) Gantry captured data

Fig. 16: Reconstruction quality of large-baseline light fields. The insets with solid boundaries show the reconstructed corner views and EPIs of different methods: **grey** denotes the ground truth, **green** denotes the direct decoding baseline, and **red** denotes ours. The white arrows indicate problematic regions.

5.5 Discussion and Limitation

Our method is the first attempt to allow light field edits through a representation learning framework. The proposed separation of visual channels and meta channel offers desirable flexibility. First, the visual channels represent the visual content of the light field as an ordinary RGB image, which can be easily edited by the well established 2D image algorithms or software. Second, the meta channel records the complimentary information and the channel number is adjustable according to the light field scale. Compared to a depth map, our meta channel is embedded with more information beyond depth and thus enables higher representation accuracy and better applicability to various scenes. In addition, our meta-view representation has a good compatibility to existing data compression techniques, which serves as another application merit.

However, due to the central-view only editing scheme, our method is restricted to support surface-only editing while fails to allow higher-order edits, such as modifying the ray-space topology or editing the material properties. Also, this limitation prevents users from editing occluded areas that are only visible from side views. Moreover, for any edits on the visual channels that affect the underlying scene geometry, the reconstructed light field tends to introduce artifacts of mismatched geometry and appearance. Fig. 17 shows two examples, where adding or removing a foreground object in the visual channels causes reconstruction distortion. Regarding this, a potential solution is to allow the meta channel updating adaptively to match the edited visual channels, possibly through promoting the decoding subnetwork with additional modules. Anyhow, it is an inherent limitation of our method that ray-space edits can never be enabled since only the central-view is available for editing.



(a) Visual channels (b) Edited visual channel (c) Reconstructed corner view

Fig. 17: Limitation illustration. (a) Object removal by image inpainting. (b) Object inserting by image composition. Both cases introduce noticeable artifacts because the visual edits no longer match the original scene geometries. The white arrows indicates the distortions.

6 Conclusion

We propose a novel representation learning framework that encodes the light field into a single compact and editable meta-view. It enables light fields to be transmitted and edited efficiently through existing 2D tools and pipelines. Experimental results show that the meta-view has high representation fidelity and good compatibility to existing compression algorithms. A series of advanced 2D image edits are effectively applied to light fields through our framework. However, our current method is restricted to the visual edits that have to preserve the original scene geometry untouched. Next step is geometry-involved edits that may require the combination of edits and updates to both visual and meta channels. Extension to light field videos is also interesting and promising. At last, our proposed edit-aware representation learning opens a new direction for light field processing and we expect it to inspire follow-up works.

Acknowledgment

This project is supported by the Research Grants Council of the Hong Kong Special Administrative Region, under RGC General Research Fund (Project No. CUHK 14201017).

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.

[3] R. Ng, M. Levoy, M. Brédif, D. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Comput. Sci. Tech. Rep.*, vol. 2, no. 1, pp. 1–11, 2005.

[4] T. Wang, J. Zhu, H. Ebi, M. Chandraker, A. A. Efros, and R. Ramamoorthi, "A 4d light-field dataset and CNN architectures for material recognition," in *European Conference on Computer Vision (ECCV)*, 2016.

[5] A. Alperovich, O. Johannsen, M. Strecke, and B. Goldluecke, "Light field intrinsics with a deep encoder-decoder network," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[6] J. Chen, J. Hou, and L. Chau, "Light field compression with disparity-guided sparse coding based on structural key views," *IEEE Trans. Image Processing (TIP)*, vol. 27, no. 1, pp. 314–324, 2018.

[7] E. Miandji, S. Hajisharif, and onas Unger, "A unified framework for compression and compressed sensing of light fields and light field videos," *ACM Trans. Graph. (TOG)*, vol. 38, no. 3, pp. 23:1–23:18, 2019.

[8] A. Jarabo, B. Masiá, A. Bousseau, F. Pellacini, and D. Gutierrez, "How do people edit light fields?" *ACM Trans. Graph. (TOG)*, vol. 33, no. 4, pp. 146:1–146:10, 2014.

[9] K. Chen, M. Chang, and Y. Chuang, "Light field image editing by 4d patch synthesis," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2015.

[10] F. Zhang, J. Wang, E. Shechtman, Z. Zhou, J. Shi, and S. Hu, "Plenopatch: Patch-based plenoptic image manipulation," *IEEE Trans. Vis. Comput. Graph. (TVCG)*, vol. 23, no. 5, pp. 1561–1573, 2017.

[11] 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, pp. 12:1–12:13, 2014.

[12] S. Vagharshakyan, R. Bregovic, and A. P. Gotchev, "Image based rendering technique via sparse representation in shearlet domain," in *IEEE International Conference on Image Processing (ICIP)*, 2015.

[13] 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, 2014.

[14] J. Li, M. Lu, and Z. Li, "Continuous depth map reconstruction from light fields," *IEEE Trans. Image Processing (TIP)*, vol. 24, no. 11, pp. 3257–3265, 2015.

[15] M. Guo, H. Zhu, G. Zhou, and Q. Wang, "Dense light field reconstruction from sparse sampling using residual network," in *Asian Conference on Computer Vision (ACCV)*, 2018.

[16] Y. Yoon, H. Jeon, D. Yoo, J. Lee, and I. S. Kweon, "Learning a deep convolutional network for light-field image super-resolution," in *IEEE International Conference on Computer Vision Workshop (ICCVW)*, 2015.

[17] N. K. Kalantari, T. Wang, and R. Ramamoorthi, "Learning-based view synthesis for light field cameras," *ACM Trans. Graph. (TOG)*, vol. 35, no. 6, pp. 193:1–193:10, 2016.

[18] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on EPI," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[19] H. W. F. Yeung, J. Hou, J. Chen, Y. Y. Chung, and X. Chen, "Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues," in *European Conference on Computer Vision (ECCV)*, 2018.

[20] L. Ni, H. Jiang, J. Cai, J. Zheng, H. Li1y, and X. Liu, "Unsupervised dense light field reconstruction with occlusion awareness," *Comput. Graph. Forum (CGF)*, vol. 38, no. 7, pp. 425–436, 2019.

[21] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and R. Ng, "Learning to synthesize a 4D RGBD light field from a single image," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[22] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 46:1–46:12, 2013.

[23] A. K. Vadathya, S. Cholleti, G. Ramajayam, V. Kanchana, and K. Mitra, "Learning light field reconstruction from a single coded image," in *Asian Conference on Pattern Recognition (ACPR)*, 2017.

[24] Y. Inagaki, Y. Kobayashi, K. Takahashi, T. Fujii, and H. Nagahara, "Learning to capture light fields through a coded aperture camera," in *European Conference on Computer Vision (ECCV)*, 2018.

[25] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, "Stereo magnification: learning view synthesis using multiplane images," *ACM Trans. Graph. (TOG)*, vol. 37, no. 4, pp. 65:1–65:12, 2018.

[26] J. Flynn, M. Broxton, P. E. Debevec, M. DuVall, G. Fyffe, R. S. Overbeck, N. Snavely, and R. Tucker, "Deepview: View synthesis with learned gradient descent," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[27] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, "Pushing the boundaries of view extrapolation with multiplane images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 175–184.

[28] C. Birklbauer and O. Bimber, "Light-field retargeting," *Comput. Graph. Forum (CGF)*, vol. 31, no. 2, pp. 295–303, 2012.

[29] L. Wang, S. Lin, S. Lee, B. Guo, and H. Shum, "Light field morphing using 2d features," *IEEE Trans. Vis. Comput. Graph. (TVCG)*, vol. 11, no. 1, pp. 25–34, 2005.

[30] M. L. Pendu, X. Jiang, and C. Guillemot, "Light field inpainting propagation via low rank matrix completion," *IEEE Trans. Image Processing (TIP)*, vol. 27, no. 4, pp. 1981–1993, 2018.

[31] S. M. Seitz and K. N. Kutulakos, "Plenoptic image editing," *International Journal of Computer Vision (IJCV)*, vol. 48, no. 2, pp. 115–129, 2002.

[32] A. Jarabo, B. Masia, and D. Gutierrez, "Efficient propagation of light field edits," in *Ibero-American Symposium in Computer Graphics (SIACG)*, 2011.

[33] S. Beigpour, S. Shekhar, M. Mansouryar, K. Myszkowski, and H.-P. Seidel, "Light-field appearance editing based on intrinsic decomposition," *Journal of Perceptual Imaging*, vol. 1, no. 1.

[34] C. Conti, P. Nunes, and L. D. Soares, "Hevc-based light field image coding with bi-predicted self-similarity compensation," in *IEEE International Conference on Multimedia Expo Workshops*, 2016.

[35] K. Marwah, G. Wetzstein, Y. Bando, and R. Raskar, "Compressive light field photography using overcomplete dictionaries and optimized projections," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 46:1–46:12, 2013.

[36] 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, 2016.

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

[38] C. Conti, P. Lino, P. Nunes, L. D. Soares, and P. L. Correia, "Spatial prediction based on self-similarity compensation for 3d holoscopic image and video coding," in *IEEE International Conference on Image Processing*, 2011.

[39] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the h. 264/avc video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, 2003.

[40] C. Perra and P. Assuncao, "High efficiency coding of light field images based on tiling and pseudo-temporal data arrangement," in *IEEE International Conference on Multimedia Expo Workshops*, 2016.

[41] J. Chen, J. Hou, and L.-P. Chau, "Light field compression with disparity-guided sparse coding based on structural key views," *IEEE Trans. Image Proc.*, vol. 27, no. 1, pp. 314–324, 2017.

[42] L. W. Dong Liu, L. Li, Z. Xiong, F. Wu, and W. Zeng, "Pseudo-sequence-based light field image compression," in *IEEE International Conference on Multimedia Expo Workshops*, 2016.

[43] 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 Journal of Selected Topics in Signal Processing*, vol. 11, no. 7, pp. 1107–1119, 2017.

[44] T. Wang, A. A. Efros, and R. Ramamoorthi, "Occlusion-aware depth estimation using light-field cameras," in *IEEE International Conference on Computer Vision (ICCV)*, 2015.

[45] M. Ruder, A. Dosovitskiy, and T. Brox, "Artistic style transfer for videos," in *German Conference on Pattern Recognition*, 2016.

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Processing (TIP)*, vol. 13, no. 4, pp. 600–612, 2004.

[47] A. S. Raj, M. Lowney, R. Shah, and G. Wetzstein, "The stanford lytro light field archive," 2016. [Online]. Available: http://lightfields.stanford.edu/LF2016.html
[48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint:1511.06349*, 2014.
[49] B. Hériard-Dubreuil, I. Viola, and T. Ebrahimi, "Light field compression using translation-assisted view estimation," in *Picture Coding Symposium (PCS), pages = 1–5, year = 2019*.
[50] D. A. Huffman, "A method for the construction of minimum-redundancy codes," in *Proceedings of the IRE*, 1952.
[51] W. Hu, M. Xia, C.-W. Fu, and T.-T. Wong, "Mononizing binocular videos," *ACM Trans. Graph. (TOG)*, vol. 39, no. 6, pp. 228:1–228:16, 2020.
[52] X. Cun, C.-M. Pun, and C. Shi, "Towards ghost-free shadow removal via dual hierarchical aggregation network and shadow matting gan," in *AAAI Conference on Artificial Intelligence (AAAI)*, 2020.
[53] J. Yoo, Y. Uh, S. Chun, B. Kang, and J. Ha, "Photorealistic style transfer via wavelet transforms," in *IEEE International Conference on Computer Vision (ICCV)*, 2019.
[54] R. Wang, Q. Zhang, C. Fu, X. Shen, W. Zheng, and J. Jia, "Underexposed photo enhancement using deep illumination estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
[55] X. Wang and J. Yu, "Learning to cartoonize using white-box cartoon representations," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[56] J. Justin, A. Alahi, and F.-F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision (ECCV)*, 2016.
[57] D. Hart, J. Greenland, and B. S. Morse, "Style transfer for light field photography," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.
[58] S. Heber and T. Pock, "Convolutional networks for shape from light field," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
[59] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.
[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

Menghan Xia received the B.Eng. degree in Remote Sensing Science and Techniques in 2014 and the Master degree in Pattern Recognition and Intelligent System in 2017 from Wuhan University, China. He is now pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. His research interests include image processing, computational photography and computer vision.

Jose Echevarria obtained his Ph.D. from University of Zaragoza, Spain, in 2016. He is currently a research scientist at Adobe Research, America. His research interest span different areas in computer graphics, computer vision and some HCI. He focus on developing novel creative tools for professional and novice users alike and turning artistic knowledge and human perception into practical solutions that people can use to explore their creativity, while developing their visual literacy.

Minshan Xie received the B.Eng. degree and Master degree in the Computer Science and Technology from South China University of Technology, China, in 2015 and 2018. She is currently working toward the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong. Her research interests include computer graphics, image processing, computer vision and deep learning.

Tien-Tsin Wong graduated from the Chinese University of Hong Kong in 1992 with a B.Sc. degree in Computer Science. He obtained his M.Phil. and Ph.D. degrees in Computer Science from the same university in 1994 and 1998 respectively. In August 1999, he joined the Computer Science & Engineering Department of the Chinese University of Hong Kong. He is currently a professor. He is a core member of Virtual Reality, Visualization and Imaging Research Centre in The Chinese University of Hong Kong. His main research interests include computer graphics, computational manga, precomputed lighting, image-based rendering, GPU techniques, medical visualization, multimedia compression, and computer vision.