A Learned Compact and Editable Light Field Representation
Menghan Xia, Jose Echevarria, Minshan Xie, and Tien-Tsin Wong

Abstract—Light fields are 4D scene representation typically structured as arrays of views, or several directional samples per pixel in a single view. This highly correlated structure is not very efficient to transmit and manipulate (especially for editing), though. To tackle these problems, we present a novel compact and editable light field representation, consisting of a set of visual channels (i.e. the central RGB view) and a complementary meta channel that encodes the residual geometric and appearance information. The visual channels in this representation can be edited using existing 2D image editing tools, before accurately reconstructing the whole edited light field back. We propose to learn this representation via an autoencoder framework, consisting of an encoder for learning the representation, and a decoder for reconstructing the light field. To handle the challenging occlusions and propagation of edits, we specifically designed an editing-aware decoding network and its associated training strategy, so that the edits to the visual channels can be consistently propagated to the whole light field upon reconstruction. Experimental results show that our proposed method outperforms related existing methods in reconstruction accuracy, and achieves visually pleasant performance in editing propagation.

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

I. INTRODUCTION

4D Light fields model incoming light rays hitting the camera sensor at different locations and from different angles, allowing them to capture complex geometries and material appearances faithfully [1], [2]. This rich representation has applications in synthesis of novel viewpoints and refocusing [3], or geometry [4] and material analysis [5]. Given a raw light field is 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. Several compression [6], [7] and editing techniques [8], [9], [10] have been proposed separately. However, although the latter enable new kinds of edits only possible with light fields, the amount of highly correlated data needed in run-time makes naive approaches inefficient, and the proposed tools are too disruptive for typical edits, with respect to the established 2D image editing workflows. So, a more memory-efficient data structure 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-view multi-channel image, 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 material properties. This meta channel would allow to reconstruct the original light field deterministically. Obviously, this representation is compact thanks to the efficient information encoding in the meta channel. Furthermore, we require the meta channel to be compatible to edits to 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 interface, which requires memory no more than a typical 2D RGBA image and supports popular 2D image processing algorithms natively (Fig. 1). We believe it makes an importance step to promote the light field consumption at scale, as currently prevented by large bandwidth and specialized complex editing tools.

To construct such a representation, we propose to learn the encoding-decoding scheme in data-driven manner. Inspired by Xia et al. [11], we adopt an autoencoder framework (Fig. 2). Specifically, the encoder subnetwork converts a light field into a single meta channel that, together with the reference
RGB view, forms the representation that allows the decoder subnetwork to recover the full light field any time. The two subnetworks are then jointly trained to facilitate the invertibility of the learned representation of the original light field. However, it is non-trivial to reconstruct a desired light field from a possibly edited representation. To facilitate the edit-aware reconstruction, we specifically decompose the decoding process into three modules: feature separation that extracts individual view information from the meta channel; disparity recovery that warps the visual channels to other views; fusion synthesis that further 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 module not only integrates the information encoded from the input, but also predicts the potential editing effects of the occlusion regions based on the surrounding context. We train our model using publicly available light field datasets to learn the invertible representation under self-supervision. Additionally, we introduce several edits to the visual channels during training, in order to enforce the editing propagation functionality.

To verify the feasibility of our approach, we applied our method on several light fields from publicly available datasets, which were captured with Lytro cameras under various settings. Extensive quantitative and qualitative evaluations were conducted. Results show that our method achieved high reconstruction accuracy and consistent editing propagation effects. In summary, our main contributions include:

- We present a novel light field representation that is both compact and editable. 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 method to learn such representation and its reconstruction, which achieves quality representation accuracy and edit propagation.
- Our method opens a promising direction for handy light field processing, which could promote light field applications to massive use.

II. RELATED WORKS

A. 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 [12], [13] or estimate the disparity to obtain them through warping [14], [15]. 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 [16], [17]. Kalantari et al. [18] synthesize a novel view with two sequential CNNs that estimate disparity for warping and refine the colors jointly. Wu et al. [19] 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. [20] employ spatial-angular alternating convolutions within a residual learning framework. Different from these methods requiring supervision of dense light fields, Ni et al. [21] 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. [22] explores the task of synthesizing a light field from a single image, achieving impressive results for very specific scenes. In all of them, the ill-posed sparse-to-dense reconstruction struggles to generate or recover information invisible in the input views due to occlusions.

A different direction for light field reconstruction uses single or few coded images [23], [24], [25], which as a compact representation at data capturing targets for accurate reconstructions. Anyhow, such coded images are usually generated via hand-crafted procedures, which significantly limits their expressiveness and accuracy; and using the visual channels for the encoding limits their editability afterwards. More recently, multiplane images (MPIs) have been also proposed as an effective representation for light fields [26], [27], [28], with very efficient generation of new views 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 seem to model the implicit geometries and materials faithfully.

B. Light Field Editing

Most previous works focus on specific editing tasks such as retargeting [29], morphing [30] or completion [31]. 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. [32] propagate local edits across multiple views through a voxel-based light field representation. Jarabo et al. [33] 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. [34] 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 propose a new compact editable light field representation that enables light fields to be edited efficiently using existing 2D image editing tools.

III. COMPACT EDITABLE LIGHT FIELDS

Given a 4D light field, we aim to represent it as a single meta channel and several visual channels (like an RGB image), which in the one hand is apparently compact, and in the
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. 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 a full light field with all the edits consistently propagated.

other, the edits on the visual channels can be propagated consistently to the rest of the views whenever the full light field is reconstructed.

A. Problem Formulation

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

$$Z = E(L)$$

$$\hat{L} = D(Z, \hat{I}_c)$$

(1)

(2)

where $E$ and $D$ roughly work as a pair of inverse functions with their own trainable parameters. According to our goal, the reconstructed $\hat{L}$ should be as similar as possible to a target light field $\hat{L}$, whose appearance is coherent with $\hat{I}_c$, while maintaining the same scene geometry of $L$. This process becomes very challenging because of (dis)occlusions, since the meta channel is encoded from the original light field, independently from the potential edits to the visual channels. As a starting point, $\hat{I}_c = I_c$ implies no editing involved, so we have $\hat{L} = L$, which enables self-supervised learning. In Section III-C, we will discuss how to train our model with edits propagation considered.

B. Representation Space

Our model learns to represent a light field as a single meta channel paired with the central RGB view. The central view captures the reference visual content of the light field, so the encoder only needs to learn a complementary residual map, i.e. the meta channel. Together, they could be encoded with all the information contained in the original light field, which is guaranteed by the autoencoder structure of the model in Fig. 2. Therefore, the meta channel plays the role of recording the original light field information complementary to the central view image, including: disparity between the central view and each other view, the occlusion content of other views from the central view, and other view-dependent information like non-Lambertian reflection. As illustrated in Figure 3, the encoded meta channel resembles a depth map of the central view, with additional texture. This implies that apart from the depth/disparity information, other information is implicitly encoded in the form of structured patterns. To verify this, we conduct an ablation study in Section V-C where the meta channel is replaced with a depth map of the central view and the reconstruction turns problematic.

Naturally, as input light field scale (e.g. spatial resolution or angular resolution) increases, the single meta channel may not have enough capacity to represent all those necessary information. In such situation, we could adaptively increase the number of meta channels. Alternatively, the whole light field even could be encoded into the visual channels $F$ directly via the encoder, i.e. $F = E(L)$, which means no meta channel is used. This is feasible to serves as a compact representation. However, the encoded visual channels are not tolerant to edits, because the implicitly encoded information may get damaged. So, separation between visual and meta channels is preferred.

Fig. 3: Representation visualization: Visual channels, i.e. the central view (a); Meta channel (b) that roughly resembles the depth distribution of the central view, with some extra structured patterns. The depth map (c) for this light field was estimated using [35].

Fig. 4: Statistical reconstruction accuracy against the meta channel numbers. This experiment is conducted over our testing dataset described in Section IV.
in our targeted scenario. In particular, for the case of 7 × 7 views in our dataset (detailed in Section IV), one meta channel achieves the best balance between efficiency and compactness, as the result shown in Fig. 4.

C. Editing-Aware Reconstruction

A U-shaped network is employed as the encoder, which learns a feature-domain representation of the input. However, it is more complex for the decoding process, which reconstructs a light field from an intermediate representation undergone potential edits. The naive solution of modeling the decoder with a single network, suffers from poor generalization to various edits to the visual channels (see Figure 12). Instead, we propose to break the decoder into several submodules: SepNet, DispNet and FusionNet, each of which learns a specific task, as the schematic diagram shown in Figure 5. The detailed architectures of the encoder and decoder are provided in the supplementary material.

Feature Separation - SepNet. As stated in Section III-B, 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, we utilize SepNet module to decompose the meta channel $Z$ into a sequence of feature maps $\{F_i\}_{i=1}^{M \times N}$, where each $F_i$ is self-contained and contains the exclusive information of view $i$. Then, each view $\hat{I}_i$ can be reconstructed from the feature map $F_i$ and visual channels $I_c$ independently.

Disparity Recovery - DispNet. We extract from $F_i$ the disparity information of view $i$, i.e. the disparity map $D(u_i)$, through DispNet. Specifically, $D(u_i, x)$ is a 2D vector, sliced from the 4D disparity field of $L$, which denotes the disparity of pixel $L(u_i, x)$ with respect to its horizontal and vertical neighboring views. Specifically, given view $L(u_i)$ and the disparity map $D(u_i)$, we can obtain the warped view $i$ via:

$$\hat{L}(u_i, x) = L(u_j, x + (u_j - u_i) \cdot D(u_i, x))$$  \hspace{1cm} (3)

So, once we have the disparity map $D(u_i)$, we can obtain the corresponding warped view by warping the visual channels $I_c$ immediately. Obviously, the warped view $\hat{L}(u_i)$ will not be exactly the same as our target view $\hat{L}(u_i)$ because of occlusions and non-Lambertian effects. Following the idea of Ruder et al. [36], we can estimate the occlusion regions by performing a forward-backward consistency check of the disparity. In particular, when warping $I_c$ to view $i$, the occlusion map $i$ can be approximated as:

$$O(u_i, x) = ||D(u_i, x) - D(u_c, x + (u_c - u_i) \cdot D(u_i, x))||_1$$  \hspace{1cm} (4)

where the greater the value of $O(u_i, x)$, the more likely the sample $L(u_i, x)$ is occluded in $\hat{I}_c$, 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 $\hat{I}_i$ by FusionNet through residual learning. Particularly, $F_i$ provides the necessary view-dependent information of the original light field, while the occlusion map $O_i$ indicates the problematic regions of the warped view $\hat{I}_i$. 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. Actually, this issue depends on the context of the occlusion regions, and we instruct the model to learn these principles from large scale training pairs, i.e. $\{\hat{I}_i, \hat{L}\}$. Whereas, it is impractical to manually prepare a diversely edited version of the light field dataset, as inter-view consistency is non-

---

Fig. 5: A schematic illustration of the decoder. The input meta channel is decomposed into sequential features $\{F_i\}_{i=1}^{M \times N}$ by the SepNet module. DispNet takes $F_i$ to generate the corresponding disparity map $D(u_i)$. A warped view $\hat{I}_i$ is generated from the disparity map and the visual channels. Then, an occlusion mask is generated by checking inter-view disparity consistency, which it is fed along with $\hat{I}_i$ and $F_i$ to FusionNet, to reconstruct the targeted view $i$. 
trivial to maintain. Instead, we propose to make use of some
global editing operators $G(\bullet)$, i.e. changing the hue, saturation,
[49x215]exposure, and contrast, whose parameters are sampled from
an uniform distribution. The nice thing is that we can easily
prepare the training pairs $\{G(I_c), G(L_1)\}$ by applying these
operators to the light field view individually, where the inter-
view consistency can be preserved well. Notably, though only
naïve 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 V-B. The
possible explanation is that these global edits and arbitrary
local variant edits share the same essence of changing pixel
values, and the model has to learn to propagate these changes
to other views. Figure 6 evidences that our model can
propagate the edited colors to those occluded areas through
this strategy.

D. Loss Function

The encoding and decoding subnetworks are jointly trained
by minimizing a loss function that combines warping consis-
tency loss $L_W$, disparity regularity loss $L_D$, and reconstruc-
tion loss $L_R$:

$$L = \omega_1 L_W + \omega_2 L_D + \omega_3 L_R$$

(5)

where $\omega_1, \omega_2, \omega_3$ are the trade-off coefficients among different
loss terms. We empirically set $\omega_1 = 0.5, \omega_2 = 0.01, \omega_3 = 1.0$
in our experiments.

First, the warping consistency loss measures the pixel-wise
difference between the light field warped from the central view
$I_c$ and the input light field as:

$$L_W = \mathbb{E}_{l_i \in S} \|I_i - L_i\|_1$$

(6)

which actually constrains 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 $\mathbb{E}_{l_i \in S}$ denotes the average operator over all the
light fields in the training dataset $S$.

Only with $L_W$, the accuracy of disparity maps are still
not guaranteed, because textureless regions are insensitive to
incorrect disparity under $L_W$. 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:

$$L_D = \mathbb{E}_{l_i \in S} \{\|D(u_i, x) - D(u_i - 1, x + D(u_i, x))\|_1\}$$

(7)

which encourages both the predicted disparity to be consistent
across views and the occluders to be sparse. For efficiency,
the disparity consistency of each view is only checked with
three neighbor views, i.e. $I = \{(0, 1), (1, 0), (1, 1)\}$. 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 $\hat{L}$ to be the
same as the target one $L$ via the reconstruction loss:

$$L_R = \mathbb{E}_{l_i \in S} (\alpha L_i - \hat{L}_i) + \beta |SSIM(L_i, \hat{L}_i)|_1$$

(8)

where SSIM($\bullet, \bullet$) measures the average SSIM [37] over all
the views of light fields. $\alpha = 1.0$ and $\beta = 0.02$ are used to
balance the magnitude between the two loss terms. We found
SSIM helps achieving more accurate details, especially in the
disoccluded regions.

IV. IMPLEMENTATION DETAILS

Dataset. Our dataset is built from two publicly available
datasets: the Stanford Lytro Lightfield Archive [38] and the
dataset from Kalantarou et al. [18], both of which were captured
with Lytro Illum cameras. In particular, 304 light fields
from [38] and 102 light fields from [18] compose our dataset,
with repeated or low-quality samples manually identified and
removed. Each light field has $376 \times 541$ spatial samples and
$14 \times 14$ angular samples, but only the central $7 \times 7$ 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 ones were randomly selected
as the training dataset and the remaining 80 were used as the
testing dataset. In the training dataset, each light field
exemplar, containing 49 RGB views, is randomly cropped
into 20 patches of $128 \times 128$ pixels. Also, we augment each
patch by flipping it horizontally or rotating it $180^\circ$, 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 computing average
gradient, we get 14,447 light field samples to train our model.

Training. As stated in Section III-D, both the encoding
subnetwork and decoding subnetwork are trained jointly.
However, we found that directly training the whole networks
from scratch suffers from low-speed convergence. This is
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
TABLE I: Reconstruction evaluation on our testing dataset (central 5 × 5 views used). Higher PSNR/SSIM is better.

| Method       | PSNR Mean | PSNR Stddev | SSIM Mean | SSIM Stddev |
|--------------|-----------|-------------|-----------|-------------|
| Inagaki et al [25] | 32.676    | 0.8508      | 0.9211    | 0.0095      |
| Ours         | 37.232    | 2.0698      | 0.9638    | 0.0083      |

Fig. 7: Visual comparison with [25]. The insets show the reconstruction details while the EPIs evaluate both the inter-view consistency and parallax correctness. The boundary color of inset represents different methods (grey denotes the ground truth), and yellow arrows indicate problematic regions.

We implemented our method using Pytorch1, a Python-based open-source deep learning framework. All the experiments were 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. The source code and trained models will be made public upon publication.

V. EXPERIMENTAL RESULTS

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

A. Reconstruction Accuracy

We evaluate the reconstruction accuracy of our representation when no edits are performed. The peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) are used as measurement metrics. As a compact light field representation that is instantly viewable, our method is firstly compared with two existing baselines: Inagaki et al. [25] represent a light field as one or few coded images by learning micro-lens weighting masks; Srinivasan et al. [22] synthesize a light field from a regular 2D image. In any case, we would like to clarify that those methods were devised for different purposes, so more than to compare with them, we use them to highlight the value of our representation.

To compare with [25], we use 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 fields for comparison. The results are presented in Table I, which shows our method outperforms [25] at a large margin. The main reason is that [25] regulates the encoding scheme as polynomial combination of the input light field views, which significantly limits the solution space. On the contrary, 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 Figure 7. It is worth noting that as encoded with information, this 2-image representation proposed in [25] cannot be directly edited.

To compare with [22], we download the source code and dataset from the author’s website and train it from scratch. Since synthesizing a light field from a single image is extremely ill-posed, the model trained on specific type of scene usually generalize poorly to new types of scenes. So for fairness, we perform comparisons over the dataset used in their paper, which consists of 100 light fields of flowers. The quantitative results are tabulated in Table II and a typical example for visual comparison is illustrated in Figure 8. As expected, [22] can not recover accurate details that are occluded in the input single image, so they are filled with dark pixels or surrounding textures instead. However, it is

1Pytorch: https://pytorch.org/

TABLE II: Reconstruction evaluation on Flowers dataset (7 × 7 views). Higher PSNR/SSIM is better.

| Method       | PSNR Mean | PSNR Stddev | SSIM Mean | SSIM Stddev |
|--------------|-----------|-------------|-----------|-------------|
| Srinivasan et al [22] | 33.365    | 3.2325      | 0.9115    | 0.0380      |
| Ours         | 40.944    | 1.8097      | 0.9719    | 0.0062      |

Fig. 8: Visual comparison with [22]. The insets show the reconstruction details while the EPIs evaluate both the inter-view consistency and parallax correctness. The boundary color of inset represents the different methods (grey denotes the ground truth), and white arrows indicate problematic regions.
fair to say that this method was not meant to be used as a compact representation for captured light fields, but a solution to synthesize plausible light fields from regular 2D images.

B. Editing Propagation

Our proposed light field representation allows edits on the visual channels, as long as they do not alter the underlying scene geometry. Although this leaves out basic edits like scaling, copying or inpainting, we believe the proposed pipeline already supports plenty of advanced and useful edits. Examples of edits supported include: global/local color manipulation (e.g. exposure, contrast, saturation, hue, etc), object texture modification, and most image processing algorithms (e.g. color transfer, denoising, dehazing, shadow removal, style transfer, etc).

Firstly, we evaluate the propagated edits by visually inspecting the inter-view consistency and parallax correctness. In Fig. 9, the light fields are reconstructed from the visual channels that are edited through global and local color adjustment with Photoshop (a), shadow removal algorithm [40], photo-realistic color transfer [41], low-lightness enhancement [42], photo cartoonization [43], and style transfer technique [44] respectively. 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 fields. By comparing the EPIs, we observe that the reconstructed light fields keep a very similar geometric structure, i.e. the parallax captured by the slopes of the color strips, to the original light fields. Besides, the insets of the corner view 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 original light fields and the edit propagated ones. Note that these light fields were edited through the visual channels, using existing 2D image editing software. This is not only much more memory-friendly in run time, but allows complex edits not possible with previous light field editing tools. Readers are recommended to check out video materials for these visual comparison. Also,
TABLE III: Edit propagation evaluation on typical examples shown in Fig. 9. 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** |

more results are available in the supplementary material.

In the absence of ground-truth edited versions of our test scenes, we evaluate the edit propagation quality quantitatively in two aspects. Firstly, as one of the supported edits, global color operations, i.e., changing the hue, saturation, exposure, and contrast, are adopted to quantitatively test our method over the whole testing dataset. In particular, 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. The average PSNR and SSIM are 36.099 and 0.9539 respectively, just slightly lower than the reconstruction accuracy without edits involved (i.e., 36.265 and 0.9592). However, global color modification is just one of the edits supported by our method. For quantitative evaluation on more diverse edits, e.g., those demonstrated in Fig. 9, we propose to utilize an indirect measurement: given a light field $L$ and its edited visual channels $\bar{L}_c$, we run the encoding and decoding recurrently for twice: $L'^{cc} = D(E(L), \bar{L}_c)$ with $\bar{L} = D(E(L), \bar{L}_c)$. Then, we can measure the difference between the recurrently reconstructed light field $L'^{cc}$ and the original light field $L$ conveniently. The results are tabulated in Table III with a control group that has $\bar{L}_c = L_c$ provided for reference. It shows that our edit reconstruction almost has the same accuracy as the reconstruction without any edit involved, which suggests the good edit propagation quality in some sense.

TABLE IV: Quantitative comparison with baselines. The reconstruction accuracy is compared over the testing dataset, while the editing propagation is compared over the edit cases shown in Fig. 9. Higher PSNR/SSIM is better.

| Method                  | Reconstruction | Edit Propagation |
|-------------------------|----------------|------------------|
|                        | PSNR | SSIM | PSNR | SSIM |
| RGB-D baseline         | 34.139 | 0.9454 | 34.541 | 0.9657 |
| Regression baseline     | 36.505 | **0.9620** | 36.143 | 0.9723 |
| Ours                    | 36.265 | 0.9592 | **37.663** | **0.9739** |

C. Ablation Study

Meta Channel. We stated the meta channel encodes all the necessary information that enables the original light field to be reconstructed deterministically. However, one could argue that the meta channel might be no more than a depth map and all the other missing information relies on the prediction by CNN. To study this issue, we constructed a baseline that replace our learned meta channel with a depth map of the central view, 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 [35]. To ensure all the other conditions unchanged, we utilize a CNN with the same architecture as our decoder, to reconstruct the light field from such RGB-D representation under the same loss function in Eq. 5. The quantitative results are tabulated in Table IV showing that the reconstruction accuracy of RGB-D is inferior to ours. However, the margin is not very large, because the narrow baseline of light field cameras avoids most inter-view occlusions. Anyhow, the qualitative results illustrated in Fig. 10 reveal the major limitation of the RGB-D baseline with respect to our proposed representation.

Firstly, the reconstruction accuracy of RGB-D highly relies on the quality of the depth map, however depth estimation from light field is challenging itself, especially in those busy background regions with fine structures. As a result, distortion and ghost effect appeared in the inset of Fig. 10(a). In theory, the depth map is unable to record the accurate depth of object boundaries due to the finite resolution, i.e., pixels of object boundary tends to be a mixture of foreground and background, and are associated with multiple depths. Therefore, it makes...
Visual channels
Corner view of RGB-D
Corner view of ours
Original visual channels
Corner view of RGB-D
Corner view of ours
Edited visual channels

Fig. 11: 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.

no sense to take an explicit depth map for information representation. In contrast, our meta channel is adaptively learned through an auto-encoder framework as a unrestricted feature map. It is flexible enough to encode any necessary information in order to reconstruct the original light field, including those occlusions, which in contrast is impossible for a depth map. In Fig. 10(b), the occluded baluster is recovered from our representation while still missing in the result of RGB-D baseline. Naturally, these limitations of RGB-D baseline retain in the editing propagation results, as demonstrated in Fig. 11.

Decoding Network. Our proposed decoding network includes three specific subnetworks. To show its necessity, we compare it with an intuitive baseline, i.e. Regression which uses a single network (the same as the encoder) to reconstruct the target light field from the generated representation directly. The quantitative results are tabulated in Table [IV]. Regression baseline achieves the highest reconstruction accuracy, because of the more flexible decoding mechanism that runs in the unrestricted feature domain completely. However, we find that it is less compatible to edits to the visual channels, especially for those with significantly changed colors or textures. Fig. 12 illustrates two typical examples. Texture replacement causes noticeable visual artifacts and parallax failure. Severe texture modification, e.g. by artistic style transfer, makes the reconstructed views almost present zero parallax, as EPIs shown in Fig. 12(b). Also, Table [IV] tabulates the quantitative results on the edit cases exampled in Fig. 9. Comprehensively, our proposed decoding network enables our representation to reconstruct artifacts-free views with edits propagated consistently.

D. Refocusing Application

One typical application of light fields is computational refocusing. Since 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. 13 demonstrates two examples, each of which presents both close and far focusing results. By comparing to the original light fields, our reconstructed light fields achieved comparable refocusing quality. In practice, our proposed technique may even benefit to artistic creation. As it is not easy to manually simulate realistic refocusing effects, our technique enable to demonstrate physically correct refocusing effects on cartoon photos, artistic photos.

E. Discussion and Limitation

Our method is the first attempt to allow light field editing based on statistical learning. The proposed separation of visual channels and meta channel offers desirable flexibility. First, the visual channels represents the visual content of the light field as a normal RGB image, which can be easily edited by the well established 2D image algorithms or softwares. Second, the meta channel records those complimentary information as an unrestricted feature map, and its channel number is adjustable according to the light field scale. Compared to a depth map,
The decoding phase. We expect more research to enrich the updating adaptively to match the edited visual channels during the scene geometry. Future work is to allow the meta channel constructed views, because of the mismatched visual contents with object on the visual channels could cause distortion to the reconstruction of light fields of different scales.

Although we already demonstrated its potential through complex and useful edits, other edits affecting the underlying geometry and materials of the scene require further study. For example, currently compositing or removing a foreground object on the visual channels could cause distortion to the reconstructed views, because of the mismatched visual contents with the scene geometry. Future work is to allow the meta channel updating adaptively to match the edited visual channels during the decoding phase. We expect more researches to enrich the updating adaptively to match the edited visual channels during the scene geometry. Future work is to allow the meta channel constructed views, because of the mismatched visual contents with object on the visual channels could cause distortion to the reconstruction of light fields of different scales.

Although we already demonstrated its potential through complex and useful edits, other edits affecting the underlying geometry and materials of the scene require further study. For example, currently compositing or removing a foreground object on the visual channels could cause distortion to the reconstructed views, because of the mismatched visual contents with the scene geometry. Future work is to allow the meta channel constructed views, because of the mismatched visual contents with object on the visual channels could cause distortion to the reconstruction of light fields of different scales.

Additionally, our meta channel can be encoded with various information beyond depth, which enables higher representation accuracy and better generalization to light fields of different scales.

Although we already demonstrated its potential through complex and useful edits, other edits affecting the underlying geometry and materials of the scene require further study. For example, currently compositing or removing a foreground object on the visual channels could cause distortion to the reconstructed views, because of the mismatched visual contents with the scene geometry. Future work is to allow the meta channel constructed views, because of the mismatched visual contents with object on the visual channels could cause distortion to the reconstruction of light fields of different scales.

VI. CONCLUSION

We presented a new compact and editable light field representation, which enables light fields to be transferred and edited robustly in existing 2D tools and pipelines effortlessly. This efficient representation is learned through a special decoding process, split into specific subnetworks that ease the training of our model. Experimental results show the effectiveness of the proposed method, both in terms of reconstruction accuracy and quality of the propagated edits, comparable to previous work. Obvious next steps are geometry-aware edits that may require a combination of edits and updates to both visual and meta channels, or extensions to light field videos. We believe this representation opens new directions for light field processing and editing, and we hope it inspires follow-up works in this area.

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. Masía, 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] M. Xia, X. Liu, and T. Wong, “Invertible grayscales,” ACM Trans. Graph. (TOG), vol. 37, no. 6, pp. 246:1–246:10, 2018.
[12] L. Shi, H. Hassanien, A. Davis, D. Katabi, and D. Durand, “Light field reconstruction using sparsity in the continuous fourier domain,” ACM Trans. Graph., vol. 34, no. 1, pp. 12:1–12:13, 2014.
[13] 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.
[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, 2014.
[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, “Light field reconstruction using deep convolutional network on EPL,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[20] 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.
[21] L. Ni, H. Jiang, J. Cai, J. Zheng, H. Lilly, and X. Liu, “Unsupervised dense light field reconstruction with occlusion awareness,” Comput. Graph. Forum (CGF), vol. 38, no. 7, pp. 425–436, 2019.
[22] P. P. Srinivasan, T. Wang, A. Steealal, 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.

Fig. 13: Synthetic refocusing comparison between the original and the edited light fields. For each example, the results of focusing at close and far distance are illustrated. The insets offer clearer inspection.
[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

[30] 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.

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

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

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

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

[35] 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.

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

[37] 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.

[38] 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

[39] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint:1511.06349*, 2014.

[40] 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.

[41] 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.

[42] 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.

[43] X. Wang and J. Yu, “Learning to cartoonize using white-box cartoon representations,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[44] 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.