Content-Aware Warping for View Synthesis

Mantang Guo, Junhui Hou, Senior Member, IEEE, Jing Jin, Hui Liu, Senior Member, IEEE, and Jiwen Lu, Senior Member, IEEE

Abstract—Existing image-based rendering methods usually adopt depth-based image warping operation to synthesize novel views. In this paper, we reason the essential limitations of the traditional warping operation to be the limited neighborhood and only distance-based interpolation weights. To this end, we propose content-aware warping, which adaptively learns the interpolation weights for pixels of a relatively large neighborhood from their contextual information via a lightweight neural network. Based on this learnable warping module, we propose a new end-to-end learning-based framework for novel view synthesis from a set of input source views, in which two additional modules, namely confidence-based blending and feature-assistant spatial refinement, are naturally proposed to handle the occlusion issue and capture the spatial correlation among pixels of the synthesized view, respectively. Besides, we also propose a weight-smoothness loss term to regularize the network. Experimental results on light field datasets with wide baselines and multi-view datasets show that the proposed method significantly outperforms state-of-the-art methods both quantitatively and visually. The source code is publicly available at [https://github.com/MantangGuo/CW4VS](https://github.com/MantangGuo/CW4VS).

Index Terms—View synthesis, light field, deep learning, image warping, depth/disparity.

I. INTRODUCTION

Novel view synthesis aims to generate views that mimic what a virtual camera would see in between two or more reference views [1], which can benefit various downstream applications, such as 3D reconstruction [2], [3], [4], [5] and virtual reality [6], [7], [8]. Over the past decades, a considerable number of view synthesis methods [9], [10], [11], [12], [13] have been proposed (see Section II for the comprehensive review). Particularly, image-based rendering (IBR) methods [9], [10], [14], [15], [16], [17] perform view synthesis by the depth-based warping operation. Generally, they first warp input source views to the novel view based on the estimated depth map, and then blend the warped images to produce the novel view. These methods mainly focus on improving the depth estimation, the blending strategy, or post-processing refinement.

Different from existing works, in this paper, we tackle the problem of novel view synthesis based on an insight that the commonly adopted warping operation confronts with natural limitations. Specifically, the traditional warping operation synthesizes pixels of a novel view by performing interpolation using a limited neighborhood from source views, and determines the interpolation weights using only distance-based functions. The reconstruction quality is thus limited because the content information around the interpolated pixels, such as texture edges, occlusion boundaries, and non-Lambertian objects, are not considered (see Section III for the detailed analysis).

To this end, we propose content-aware warping to replace the traditional warping operation, in which a lightweight neural network is utilized to learn content-adaptive interpolation weights for pixels of a relatively large neighborhood from their contextual information. Based on this learnable warping module, we construct a new end-to-end learning-based framework for novel view synthesis from a set of input source views. To be specific, we first warp the source views separately to the novel view pixel-by-pixel via the proposed content-aware warping, and then adaptively leverage the warped views via confidence-based blending to handle the occlusion problem, leading to an intermediate result of the novel view. As the pixels of the intermediate result are independently synthesized, we subsequently recover the spatial correlation among them by referring to that of source views using the feature-assistant spatial refinement module. We also regularize the network through a weight-smoothness loss.

In summary, the main contributions of this paper are as follows:
- we analyze the classic 2D image warping operation when used for novel view synthesis and figure out its limitations in terms of the limited neighborhood and only distance-based interpolation weights;
- we propose learnable content-aware warping, which is capable of overcoming the limitations of the traditional warping operation; and
- we propose a new end-to-end learning-based framework for novel view synthesis.
A preliminary version of this work was published in ICCV’21 [18]. Compared with the conference version, the additional technical contributions of this paper are four-fold:

- we embed the global content information for learning the content-adaptive weight;
- we propose a feature-assistant spatial refinement module;
- we propose a novel weight-smoothness loss term; and
- we extend the framework to synthesize novel views on multi-view datasets and experimentally demonstrate its significant advantages over state-of-the-art approaches.

Extensive experiments on both light field (LF) and multi-view benchmark datasets demonstrate the significant superiority of our method over state-of-the-art methods. Especially, our proposed method employed to LF reconstruction task improves the PSNR of the conference version by around 1.0 dB.

The rest of this paper is organized as follows. Section II reviews related works. In Section III, we analyze the drawbacks of the traditional image warping operation when used for view synthesis and propose content-aware warping. In Section IV, we present the proposed view synthesis framework. In Section V, we conduct extensive experiments and analysis to evaluate our framework on both LF and multi-view datasets and discuss the limitation of our method. Finally, Section VI concludes this paper.

II. RELATED WORK

Based on the degree of the geometry constraint among views, we roughly divide existing view synthesis methods into two categories: LF-based view synthesis and multi-view-based view synthesis.

A. LF-Based View Synthesis

Existing LF reconstruction methods could be roughly divided into two categories: non-learning-based methods and learning-based methods.

Non-learning-based methods usually solve this inverse problem by regularizing the LF data based on different prior assumptions, e.g., Gaussian-based priors [19], [20], [21], sparse priors [22], [23], [24], and low-rank [25]. These methods either require many sparse samplings, or have high computational complexity. Another kind of methods for LF reconstruction is explicitly estimating the scene depth information, and then using it to warp input sub-aperture images (SAIs) to synthesize novel ones. Wanner and Goldluecke [26] estimated disparity maps at input view by calculating the structure tensor of epipolar plane images (EPIs), and then used the estimated disparity maps to warp input SAIs to the novel viewpoints. This method makes the reconstruction quality rely heavily on the accuracy of the depth estimation. Zhang et al. [27] proposed a disparity-assisted phase-based method that can iteratively refine the disparity map to minimize the phase difference between the warped novel SAI and the input SAI. However, the angular positions of synthesized SAIs are restricted to the neighborhood of input views, which cannot reconstruct LFs from the input with large baselines.

Recently, many deep learning-based methods have been proposed to reconstruct dense LFs from sparse samplings. Yoon et al. [28] reconstructed novel SAIs from spatially up-sampled horizontal, vertical and surrounding SAI-pairs by using three separate networks. This method can only regress novel SAIs from adjacent ones, and could not process sparse LFs with large disparities. Wu et al. [29] used a 2-D image super-resolution network to recover high-frequency details along the angular dimension of the interpolated EPI. Analogously, Wang et al. [30] restored the high-frequency details of EPI stacks by using 3-D convolutional neural networks (CNNs). These methods process 2-D or 3-D slices of a 4-D sparse LF, which cannot fully explore the spatial-angular correlations implied in the LF. Yeung et al. [31] proposed an end-to-end network to reconstruct a dense LF from a sparse one in a single forward pass, which uses the spatial-angular separable convolution to explore the 4-D angular information effectively and efficiently. Meng et al. [32] directly employed the 4-D convolution to model the high-dimensional distribution of the LF data, and proposed a coarse-to-fine framework for spatial and angular super-resolution. However, these regression-based methods always suffer from blurry effects or artifacts when input SAIs have relatively large baselines.

To handle sparse LFs with large baselines, some learning-based methods also employ the pipeline of warping-based view synthesis. Kalantari et al. [33] used two sequential networks to separately estimate the disparity map at the novel view, and predicted the color of novel SAI from warped images, respectively. Wu et al. [34] extracted depth information from the sheared EPI volume, and then used it to reconstruct high angular-resolution EPIs. These methods either ignore the angular relations between synthesized SAIs, or underuse the spatial information of the input SAIs during the reconstruction. Srinivasan et al. [35] reconstructed an LF from a single 2-D image with predicting 4-D ray depths. This method only works on dataset with small disparities, and is restricted by its generalization ability. Jin et al. [36] explicitly learned the disparity map at the novel viewpoint from input SAIs. They synthesized the coarse novel SAIs individually by fusing the warped input SAIs with confidence maps. Then they used a refinement network to recover the parallax structure by exploring the complementary information from the coarse LF.

B. Multi-View-Based View Synthesis

To synthesize novel views from a set of views, early IBR methods typically blend corresponding pixels from source views. These methods focus on improving the recovery of the scene geometry [9], [14], [15] or modulating the blending weights [10], [37], e.g., Penner and Zhang [37] constructed a soft volume for each source view, where each voxel encodes a surface/free space confidence value, and then generated a novel view by performing back-to-front synthesis, based on the ray visibility and occlusion modeled from soft volumes. Recently, many deep learning-based methods also adopt the similar scheme and aim at improving the depth estimation, source-to-target blending, and post-processing refinement. Specifically, Hedman et al. [38] first generated per-source view 3D meshes by leveraging advantages of two MVS reconstruction methods and using their proposed mesh simplification method. Then, they employed a CNN with
four source-view mosaics and a rendered novel view image from the global mesh as inputs to predict blending weights, which are finally used to calculate the weighted sum of the five inputs as a novel view. Choi et al. [39] estimated the depth map of the novel view by warping and fusing source depth probability volumes, and used the depth map to synthesize an intermediate novel view by warping and blending. They also employed a patch-based refinement module to further recover the image details for the intermediate result. Riegler and Koltun [16] obtained the novel view depth from the surface mesh. Based on the depth map, they warped the encoded source feature maps to the novel view, and then blended them via another network to further decode the novel view image. Shi et al. [17] warped source view feature maps to the target view to construct a feature frustum, and then compared their similarity to estimate the source visibility as well as the target depth probability, which are further used to warp and aggregate the source views to produce the novel view.

More recently, some learning-based methods for view synthesis propose to introduce specific scene representations with corresponding differentiable rendering procedures, e.g., MPI [11], [12], [40], [41], [42], and NeRF [13], [43], [44]. MPI is composed of a set of fronto-parallel planes at discrete depths, and each plane consists of an RGB image and an alpha image to encode the color and visibility information at the current depth. The novel view can be rendered from the MPI by compositing the color images in back-to-front order using the differentiable over operation [45]. Zhou et al. [11] predicted the MPI at a reference view by using a CNN to represent the scene’s content. Then the novel view can be synthesized from the MPI representation with homography and alpha compositing. Flynn et al. [40] estimated an initial MPI from the source views, and then iteratively improved the initial MPI via learned updates which incorporate visibility information to improve the performance. Mildenhall et al. [12] first independently expanded each source view to a local MPI, and then rendered the novel view by fusing adjacent local MPIs. NeRF [43] use a multilayer perceptron (MLP) as an implicit function to represent the continuous radiance field, which outputs the view-dependent color and volume density at each 3-D spatial position. Then the novel view can be rendered through the differentiable and classical volume rendering [46]. Afterwards, Yu et al. [44] and Wang et al. [13] proposed to input image feature together with spatial coordinate and view direction to the MLP so as to learn generic NeRF functions that can generalize to other unseen scenes.

III. ANALYSIS OF CLASSIC 2D IMAGE WARping

Given $N$ input source views denoted by $\{I_i\}_{s=1}^N$, we aim to synthesize the unsampled novel view between them, denoted by $\tilde{I}_t$, which should be as close to the ground-truth $\tilde{I}_t$ as possible.

Let $I_t(x_t)$ and $I_s(x'_t)$ be the projections of a typical scene point in different views, where $x_t$ and $x'_t$ are the 2D spatial coordinates of the pixels, and under the assumption of Lambertian, we have

$$I_t(x_t) = I_s(x'_t).$$

(1)

Moreover, without occlusions, the relation between $x_t$ and $x'_t$ can be computed as

$$x'_t = K_s(R_sR_t^\top dK_t^{-1}x_t + t_s - R_sR_t^\top t_t),$$

(2)

where $d$ is the depth value of pixel $I_t(x_t)$, and $K_s$, $R_s$, and $t_s$ (resp. $K_t$, $R_t$, and $t_t$) are the intrinsic matrix, the rotation matrix, and the translation vector of the source (resp. novel) view, respectively. Thus, to synthesize $\tilde{I}_t$, for each pixel position of $\tilde{I}_t$, one can figure out the corresponding position in $I_t$, according to (2), and then map its pixel value to $\tilde{I}_t$. However, as the values of $x'_t$ are not always integers, interpolation has to be performed to compute the intensity of the corresponding pixel, and the process can be formulated as

$$\tilde{I}_t(x_t) = \sum_{x'_s \in P_{x'_t}} w(x_s - x'_t; \phi_w)I_s(x_s),$$

(3)

where $P_{x'_t}$ is the set of 2D coordinates of the pixels neighbouring to $x'_t$, and function $w(\cdot; \cdot)$ with the parameter $\phi_w$ defines the interpolation weights for the pixels of $I_s$, based on the distance between two pixels.

The above mentioned procedure is the image backward warping operation widely-used in view synthesis. However, we argue that this procedure has natural limitations. Specifically, as shown in Fig. 1(a), this process pre-defines the neighbors used for synthesizing the target pixel based on the adopted interpolation kernel, e.g., $2 \times 2$ pixels around the corresponding point are

![Image](https://example.com/image.png)

**Fig. 1.** Comparison of the traditional image warping operation and the proposed content-aware warping. In contrast to the content-independent weights employed in the warping operation (taking bilinear interpolation weights as an example), we propose to learn geometry-aware and content-adaptive interpolation weights from carefully constructed embeddings.

\[\text{We do not distinguish between pixel coordinate and homogeneous coordinate for simplification.}\]
selected as the neighbors for bilinear interpolation [47]. Besides, the weight of each neighbor is only the function of distance from the corresponding point. Thus, it is difficult to produce high-quality results, especially on areas with texture edges, occlusion boundaries, and non-Lambertian objects.

To overcome the limitations of the traditional warping operation, as shown in Fig. 1(b), we propose content-aware warping, which adaptively learns a weight value for each pixel of a relatively large interpolation neighborhood, based on the contextual information. See Section IV.A for the detailed process. We expect that such a process is able to assign a large weight to the neighbor, which is prone to the correspondence of the target pixel or semantically close to the target pixel, to emphasize its contribution, but a small one close to zero to the neighbor with a low probability of being the correspondence to exclude its interference. Based on this learnable warping process, we construct a new view synthesis framework explained in the next section.

Bako et al. [48] denoised the Monte Carlo rendering by separately filtering its diffusion and specular components. For each component, they fed the block of pre-processed color channel and auxiliary features centered at each pixel into a CNN to learn regular 2D kernels, which are further applied to the noisy color channel. Generally, we share the same motivation as Bako et al. [48], i.e., overcoming the natural limitations of pre-defined and fixed kernels. However, as we focus on the problem of novel view synthesis, the proposed method is technically very different from that in Bako et al. [48].

IV. PROPOSED METHOD

Overview. As shown in Fig. 2, the proposed view synthesis framework mainly consists of three modules, i.e., content-aware warping, confidence-based blending, and feature-assistant spatial refinement. We refer the readers to Fig. 4 for more details of the feature-assistant spatial refinement module.

Fig. 2. Flowchart of the proposed framework for view synthesis from \( N \) input source views. It consists of three modules: content-aware warping, confidence-based blending, and feature-assistant spatial refinement. We refer the readers to Fig. 4 for more details of the feature-assistant spatial refinement module.
angular code $E_{\text{ang}}^{\text{ng}}$. Specifically, $E_{\text{geo}, x_t}$ is the concatenation of disparity values of $I_t(x_t)$:

$$E_{\text{geo}, x_t} = \text{CAT} \left( \{D_{s,i}(x_t)\}_{i=1,i\neq s} \right),$$

where $D_{s,i}$ is the disparity map of $I_s$ calculated from $I_s$ and another one of the remaining source views $I_i$ by applying an off-the-shelf disparity/optical flow estimation method. In this paper, we adopt the pre-trained RAFT [49] to estimate $D_{s,i}$. $E_{\text{spa}, x_t}$ describes the spatial distance between $I_t(x_t)$ and $I_s(x_s)$, defined as

$$E_{\text{spa}, x_t} = x_s - x_t.$$  

$E_{\text{ctt}, x_t}$ describes the relative camera pose between $I_t$ and $I_s$. A simple way to embed camera pose into the MLP is to vectorize the rotation and translation matrix and concatenating their entries with other codes, which dramatically increases the number of MLP parameters. To tackle this problem, we employ the 6DoF vector [50] formed by 3-dimension translation vector and 3-dimension Euler angles to represent the camera pose. We define $E_{\text{ctt}, x_t}$ as the difference of the 6DoF camera poses between $I_t(x_t)$ and $I_s(x_s)$, i.e.,

$$E_{\text{ctt}, x_t} = z_s - z_t,$$

where $z_s$ and $z_t$ are the 6DoF camera poses of $I_s$ and $I_t$, respectively.

Given the disparities between $I_s$ and other source views, we expect the network could implicitly infer the correspondence information between $I_s$ and $I_i$. Such correspondence relation between $I_i$ and $I_s$ together with the spatial code and angular code, would be an indicator for the MLP to determine whether the neighbor corresponds to the target pixel $I_t(x_t)$, and assigns appropriate weight values to the neighbor.

2. The content information around $I_t(x_t)$. This kind of information helps to understand complicated scenarios, such as texture edges, occlusion boundaries, and non-Lambertian objects. To construct the content embedding, denoted by $E_{\text{ctt}, x_t}$, we first separately backward warp the remaining input views $\{I_i\}_{i=1,i\neq s}^N$ to $I_s$, based on $\{D_{s,i}\}_{i=1,i\neq s}^N$ generating $\{\tilde{I}_{s,i}\}_{i=1,i\neq s}^N$. Then, we employ a sub-CNN $f_c(\cdot)$ to learn the content information, i.e.,

$$E_{\text{ctt}, x_t} = f_c \left( x_s, x_t, I_s, \{\tilde{I}_{s,i}\}_{i=1,i\neq s}^N, \{D_{s,i}\}_{i=1,i\neq s}^N \right).$$

It is expected that $f_c(\cdot)$ is able to detect the texture edges of $I_s$ and understand the occlusion and non-Lambertian relations by comparing $I_s$ and $\{\tilde{I}_{s,i}\}_{i=1,i\neq s}^N$ with the assistance of $\{D_{s,i}\}_{i=1,i\neq s}^N$. In spite of per-pixel content information, we explicitly calculate the global mean $\mu_{x_t}$ and variance $\nu_{x_t}$ of $E_{\text{ctt}, x_t}$ over the neighborhood $\mathcal{P}_{x_t}$:

$$\mu_{x_t} = \frac{1}{|\mathcal{P}_{x_t}|} \sum_{x_s \in \mathcal{P}_{x_t}} E_{\text{ctt}, x_s},$$

$$\nu_{x_t} = \frac{1}{|\mathcal{P}_{x_t}|} \sum_{x_s \in \mathcal{P}_{x_t}} \left( E_{\text{ctt}, x_s} - \mu_{x_t} \right)^2,$$

where $|\mathcal{P}_{x_t}|$ is the number of pixels in $\mathcal{P}_{x_t}$. The global mean and variance can help the MLP to distinguish which neighborhoods a typical source pixel belongs to, and assign the source pixel a appropriate weight to improve the reconstruction quality. We finally construct the geometry and content embedding $E_{x_t, x_s}$ as

$$E_{x_t, x_s} = \text{CAT} \left( E_{\text{geo}, x_t, x_s}, E_{\text{ctt}, x_t, x_s}, E_{\text{ctt}, x_t}, \mu_{x_t}, \nu_{x_t} \right),$$

where $\text{CAT}(\cdot)$ is the concatenation operation, and predict the interpolation weights $W_{x_t, x_s}$ as

$$W_{x_t, x_s} = f_w \left( E_{x_t, x_s} \right),$$

B. Confidence-Based Blending

Although the content-aware warping module has the ability of handling occlusion boundaries by embedding the contextual information, it is still difficult to synthesize the pixels whose correspondences are completely occluded in a source view by only warping that source view. Considering that the object occluded from one viewpoint might be visible from other ones, we thus blend the views separately warped from $\{I_i\}_{i=1}^N$ under the guidance of their confidence maps, which indicate the non-occlusion pixels with higher values.

To predict the confidence value for each pixel position $x_t$ in the synthesized view, we first aggregate the geometry and content embeddings of all neighbors corresponding to $x_t$ in $I_s$ by the content-aware warping, i.e.,

$$\tilde{I}_{s \rightarrow t}(x_t) = \sum_{x_s \in \mathcal{P}_{x_t}} W_{x_s, x_t} E_{x_s, x_t},$$

where $\tilde{I}_{s \rightarrow t}(x_t)$ is the aggregated content and geometry embedding corresponding to $x_t$ from $I_s$. We then apply another MLP, denoted by $f_b(\cdot)$, on the concatenation of the aggregated embeddings from all source views to predict confidence values corresponding to them, i.e.,

$$\hat{C}_t(x_t) = f_b \left( \text{CAT} \left( \{ \tilde{I}_{s \rightarrow t}(x_t) \}_{s=1}^N \right) \right),$$

where $\hat{C}_t \in \mathbb{R}^{H \times W}$ is the confidence map volume formed by the predicted confidence maps for $\{I_{s \rightarrow t}\}_{s=1}^N$, and let $C_{s,t} \in \mathbb{R}^{H \times W}$ be the s-th slice, i.e., the confidence map for $I_{s \rightarrow t}$, where $H$ and $W$ are the spatial dimensions of the input source views. Based on the learned confidence maps, we then blend the warped views to produce the intermediate result of the novel view as

$$\tilde{I}_t = \sum_{s=1}^N C_{s,t} \odot \tilde{I}_{s \rightarrow t},$$

where $\odot$ is the element-wise multiplication operator.

As an example, Fig. 3 visually illustrates the advantage of such a confidence-based blending module.
C. Feature-Assistant Spatial Refinement

As the pixels of the intermediate result $\tilde{I}_t^b$ are independently synthesized, the spatial correlation among them is not taken into account, i.e., the intensities of pixels that are spatially close are generally similar. Thus, as shown in Fig. 4, we propose to propagate the spatial correlation in $I_s$ to the corresponding region of $\tilde{I}_t^b$ in both image and feature space to further refine the quality of $\tilde{I}_t^b$. Generally, in image space, we warp $I_s$ to the target view via constructing a plane-sweep volume (PSV) \cite{51}, then adaptively aggregates useful information from the warped source images at different depth layers in the constructed PSV. In feature space, we align the feature map extracted from $I_s$ to the novel view via the proposed content-aware warping. The feature-space warping is based on the observation that a typical pixel of the source view mapped to the feature space encodes the information of its local region containing the spatial correlation around it. We learn an additive map for $\tilde{I}_t^b$ by using a sub-CNN with the concatenation of the following information: $\tilde{I}_t^b$, the aggregated PSV feature maps and the content-aware-warping aligned feature maps from $\{I_s\}_{s=1}^N$.

Specifically, let $V_s \in \mathbb{R}^{D \times H \times W \times C}$ be the PSV constructed from $I_s$ with uniformly sampling $D$ depth layers within the depth range of the target view, and $V_d^s \in \mathbb{R}^{H \times W \times C}$ the warped image at $d$-th ($1 \leq d \leq D$) depth layer, where $C$ is the channel number of the source views. To adaptively aggregate the information across depth layers, we extract a feature map as well as predict a corresponding weight value at a typical depth layer $d$ by employing a sub-CNN $f_v(\cdot)$ with the concatenation of $\tilde{I}_t^b$ and $V_d^s$ as input, i.e.,

$$F_d^s, w_d^s = f_v \left( \text{CAT} \left( \tilde{I}_t^b, V_d^s \right) \right),$$

where $F_d^s \in \mathbb{R}^{H \times W \times 64}$ is the feature map corresponding to $V_d^s$, and $w_d^s$ is the weight value. $f_v(\cdot)$ estimates a 65-channel output where first 64-channel is $F_d^s$ and the last channel is for the weight. We average the values of the last channel as the weight value $w_d^s$, and normalize the weight values from different depth layers by a
softmax function. We then fuse the feature maps from different
depth layers as
\[ F_s = \sum_{d=1}^{D} a_s^d F_s^d, \]  
\[ \text{(16)} \]
where \( F_s \) is the fused PSV feature map from \( I_s \).

Furthermore, we map \( I_s \) to the feature space by employing a
sub-CNN \( f_g(\cdot) \), i.e.,
\[ G_s = f_g(I_s), \]  
\[ \text{(17)} \]
where \( G_s \) is the feature map of \( I_s \). Then, we align \( G_s \) with the
feature map of the novel view via the content-aware warping with
the learned weights in Section IV.A to perceive the corre-
spending information from \( G_s \), i.e.,
\[ \tilde{G}_{s \rightarrow t}(x_t) = \sum_{x_s \in P_{x_t}} W_{x_t,x_s} G_s(x_s). \]  
\[ \text{(18)} \]
We finally refine \( \tilde{I}_t \) as
\[ \tilde{I}_t = f_r \left( CAT \left( \tilde{I}_t, \{ F_s \}_{s=1}^{N}, \{ G_s \}_{s=1}^{N} \right) \right) + \tilde{I}_t, \]  
\[ \text{(19)} \]
where \( f_r(\cdot) \) denotes a sub-CNN.

D. Loss Function

To train the network, basically we use the ground-truth novel
view \( I_t \) to supervise both the final and intermediate predictions
of the novel view, i.e.,
\[ \ell_t^p = \left\| \tilde{I}_t - I_t \right\|_1 + \left\| \tilde{I}_t - I_t \right\|_1 + \sum_{s=1}^{N} \left\| \tilde{I}_{s \rightarrow t} - I_t \right\|_1, \]  
\[ \text{(20)} \]
where \( \left\| \cdot \right\|_1 \) denotes the \( L_1 \) norm. Besides, we also employ
the perceptual loss and ssim loss to further regularize the final prediction, which are defined as
\[ \ell_t^p = \left\| \phi(\tilde{I}_t) - \phi(I_t) \right\|_1, \]  
\[ \text{(21)} \]
and
\[ \ell_t^w = \frac{1 - SSIM(\tilde{I}_t, I_t)}{2}, \]  
\[ \text{(22)} \]
respectively, where \( \phi(\cdot) \) is the output of the layer ‘conv1_2’
of a pretrained VGG-19 network [52], and \( SSIM(\cdot, \cdot) \) is the
structural similarity index measure [53] between two images.

Moreover, spatially close pixels generally tend to have similar
intensity, and their neighborhoods used for interpolation are
mostly overlapping. Thus, the distributions of learned weights
should be similar. Based on this observation, we propose a
weight-smoothness loss term to regularize the network. Specifi-
cally, let \( \mathcal{W}_s \in \mathbb{R}^{|P_{x_t}| \times H \times W} \) be the weight map volume formed
by the learned weights from \( I_t \), for synthesizing all pixels of \( I_t \),
and \( \mathcal{W}_s^l \in \mathbb{R}^{H \times W} \) the l-th (\( 1 \leq l \leq |P_{x_t}| \)) slice. Accordingly,
the above-mentioned observation indicates that \( \mathcal{W}_s \) should
be locally constant. Therefore, we mathematically formulate the
weight-smoothness loss as
\[ \ell_t^w = \sum_{s=1}^{N} \sum_{l=1}^{|P_{x_t}|} \left( \left\| \nabla_x \mathcal{W}_s^l \right\|_1 + \left\| \nabla_y \mathcal{W}_s^l \right\|_1 \right), \]  
\[ \text{(23)} \]
where \( \nabla_x \) and \( \nabla_y \) are the gradient operators for the spatial
domain.

In all, we train the proposed framework end-to-end with the
following loss function:
\[ \ell = \ell_t^p + \ell_t^w + \lambda \ell_t^w, \]  
\[ \text{(24)} \]
where \( \lambda \geq 0 \) is the hyper-parameter to balance the two terms.

Remark. Compared with the disparity-oriented loss of the
preliminary work [54], the weight-smoothness loss is more prac-
tical because the disparity-oriented loss requires ground-truth
disparity maps, which are unavailable for real-world LF data,
while the weight-smoothness loss does not require these data.

V. EXPERIMENTS

A. Implementation Details and Datasets

The content embedding network \( f_c(\cdot) \) and the spatial refine-
ment network \( f_r(\cdot) \) are all 2-D CNNs composed of residual
blocks [58] with the kernel of size \( 3 \times 3 \). We utilized zero-
padding to keep the spatial size unchanged. The context extrac-
tion network \( f_g(\cdot) \) is a U-Net architecture which is the same as
the image encoding CNN of [59]. We refer readers to the Supple-
mentary Material, which can be found on the Computer Society
Digital Library at http://doi.ieeecomputersociety.org/10.1109/
TPAMI.2023.3242709, for the detailed network architecture. At
each iteration of the training phase, we synthesized a fixed-size
patch randomly cropped from the target image. The batch size
was set to 4. The learning rate was initially set to \( 1 \times 10^{-4} \)
and reduced to \( 1 \times 10^{-5} \) after 8000 epochs. We used Adam [60] with
\( \beta_1 = 0.9 \) and \( \beta_2 = 0.999 \) as the optimizer.

We trained and tested our network on both LF and multi-view
datasets. For LF datasets, we reconstructed the 3-D LF contain-
ing 5 SAIs, i.e., inputting SAIs at two ends as source views
to reconstruct middle three ones. Specifically, we trained our
network with 29 LF images from the Inria Sparse LF dataset [55].
Each LF image is 4-D and contains 9 \( \times \) 9 SAIs with a disparity
range of \([-20, 20] \) between adjacent SAIs. We extracted 3-D
LFS as training samples from the 4-D LF image by taking the
3rd to 7th SAIs at each row. Note that the disparity between
two input source views at each 3-D LF is up to 80 pixels.
The test dataset consists of 7 LF images from the Inria Sparse
LF dataset [55]. For each LF image, we took the 3rd and 7th
SAIs at the 5th row as input source views to reconstruct the
3-D LF. We also tested on 14 LF images from the MPI LF
archive [61]. Note that MPI [61] is a high angular-resolution
LF dataset where each LF image contains 101 SAIs distributed
on a scanline. Thus, we can construct testing LFS with different
disparities by sampling SAIs with different intervals (see details
in Section V.B). For multi-view datasets, we trained and tested
our network on both DTU[62] and RealEstate10K[11] datasets.
For DTU[62] dataset, we used the dataset preprocessed by [4],
and trained our network on 79 scenes, and tested on 18 scenes.
For RealEstate10K[11] dataset, we trained our network on 85
scenes, and tested on 17 scenes. Given multiple views of a scene,
we constructed a training sample by first randomly selecting a
novel view, and then sampling its two adjacent ones as source
views. Since different datasets have different disparity ranges,
we set various interpolation neighborhood sizes in the source view for different datasets. We refer readers to the Supplementary Material, available online, for the detailed construction of the interpolation neighborhood.

B. Evaluation on LF Datasets

We compared the proposed method with five state-of-the-art deep learning-based LF reconstruction methods, including Kalantari et al. [33], Wu et al. [34], Wu et al. [56], Jin et al. [36],2 and Guo et al. [18]. Besides, we also compared with a warping-based video frame interpolation method, i.e., Bao et al. [57] which also employs a learned warping layer. For a fair comparison, we trained all the methods on the same dataset with the officially released codes and suggested configurations. Since the disparity between source views, i.e., source-to-source disparity, and the disparity between the source and target views, i.e., source-to-target disparity, are proportional in the LF, we can easily calculate the source-to-target disparity from the source-to-source disparity. To let the MLP directly perceive correspondence relation between the source view and the target view, we used the source-to-target disparity instead of the source-to-source disparity to construct the geometry code and learn the content information in our method on the LF datasets.

To directly verify the advantage of the proposed content-aware warping over the traditional warping operation, we constructed a baseline model, named Baseline (Warp), by replacing the content-aware warping module of our framework with a disparity-based warping operation while leaving other modules unchanged. Specifically, Baseline (Warp) first forward warps the disparity maps \( D_{1,2} \) and \( D_{2,1} \) to the novel view, and then employs a sub-CNN to predict the disparity map of the novel view and two confidence maps corresponding to the two input views from the concatenation of the warped disparity maps. Baseline (Warp) further backward warps the two input SAIs to the novel view separately based on the predicted disparity map, and blends them based on the confidence maps. Finally, the blended SAI is refined by the feature-assistant spatial refinement module.

Besides, we also set another baseline, named Baseline (Disparity), by incorporating the estimated disparity maps used in our framework into Jin et al. [36],3 so that both our method and Baseline (Disparity) perceive the same input data to achieve a fair comparison. To be specific, we modified Jin et al. [36] by adding a sub-CNN to estimate the disparity map of the novel view from the input disparity maps, and then blending the estimated target disparity map with the one estimated from the plane plane-sweep volumes (PSVs) with confidence maps.

1) Quantitative Comparisons on the Inria Sparse LF Dataset: Table I lists the quantitative comparison of different methods on the Inria Sparse dataset, where it can be observed that:

- our method improves the average PSNR of the preliminary version, i.e., Guo et al. [18], by more than 0.8 dB, which is credited to the newly proposed components, i.e., the global content information, the adaptive PSV fusion, the feature-space warping, and the weight-smoothness loss term. See Section V.D for the detailed ablation studies on these components;
- our method achieves significantly higher PSNR and SSIM than Baseline (Warp), which directly verifies the advantage of the proposed content-aware warping over the traditional warping operation;
- our method achieves higher performance than both Wu et al. [34] and Wu et al. [56]. The reason may be that they perform reconstruction on 2-D EPIs without sufficient modeling of the spatial domain of each view, while our method employs a feature-assistant refinement module to refine the spatial correlation among pixels of novel views;
- our method achieves higher PSNR and SSIM than both Kalantari et al. [33] and Jin et al. [36]. The main reason may be that in addition to the natural limitations of the adopted traditional warping operation, they also cannot provide effective refinement on LFs with large disparities. However, our method overcomes the limitations by the content-aware warping and effectively propagates the spatial correlation to the novel view via the feature-assistant refinement module. Besides, our method achieves higher performance than Baseline (Disparity), demonstrating that the advantage of our framework does not completely come from adopting more accurate disparity maps; and
- our method achieves higher performance than Bao et al. [57] on all scenes except Flowers. Although Bao et al. [57] also proposed a learned warping layer, it has to first explicitly specify the center of the interpolation neighborhood based on the optical flow and depth before conducting the interpolation. The synthesis quality would highly depend on the accuracy of the optical flow and depth estimation. Conversely, our method embeds the estimated optical flow into the MLP as the geometry code, which can tolerate the error of the optical flow estimation to some extent.

2) Quantitative Comparisons on the MPI LF Dataset: We also evaluated different methods under different disparity ranges on the MPI dataset [61]. Each scene contains a high-angular densely-sampled LF image composed of 101 SAIs distributed on a scanline with spatial resolution \( 720 \times 960 \). The disparity between adjacent SAI is around 1 pixel. We sampled SAIs with different intervals along the angular dimension to construct LFs with different disparity ranges. Specifically, we separately set 10 disparity ranges from 8 to 80 pixels between two input SAIs. For each disparity range, we evenly sampled 3 SAIs as ground truth. From Fig. 5, it can be seen that the performance of all methods decreases with the disparity range increasing because the reconstruction problem is more challenging, but our method consistently achieves the highest

---

2Note that the 2-D angular convolutional layers were degenerated to 1-D convolutional layers to adapt to the 3-D LFs.

3Here we selected Jin et al. [36] to construct Baseline (Disparity) based on the following facts. Among the compared LF reconstruction methods, Wu et al. [34] and Wu et al. [56] are both EPI-based methods which do not accept such disparity information. Kalantari et al. [33] and Jin et al. [36] are both warping-based methods which need such information, and Jin et al. [36] can achieve higher performance than Kalantari et al. [33].
We quantitatively compared our method achieves higher average PSNR and SSIM, but slower. Our method is able to reconstruct novel views with much higher quality, there is still room to improve its efficiency. Although our method is able to reconstruct novel views with much higher quality, there is still room to improve its efficiency. Although our method is able to reconstruct novel views with much higher quality, there is still room to improve its efficiency.

PSNR and SSIM among all methods under each disparity range, demonstrating the robustness of our method towards different disparity ranges.

3) Comparisons of Visual Results: We visually compared the reconstructed LFs by different methods in Figs. 6 and 7, where it can be observed that our method can produce views with sharp edges at the occlusion boundaries that are closer to ground truth, while the other methods produce views with either severe distortions or heavy blurry effects at these regions. Besides, our method can produce better high-frequency details at regions with rich textures than other methods. We refer the readers to the Supplementary Material, available online, for more visual results.

4) Comparisons of the LF Parallax Structure: As the parallax structure is one of the most important values of LF data, we thus managed to compare the parallax structures of LFs reconstructed by different methods. First, as shown in Figs. 6 and 7, for the formed EPIs, our method can preserve clearer linear structures than other methods, even for lines corresponding to regions with large disparities, demonstrating the strong ability of our method in preserving the parallax structure on extremely sparse LFs. Generally, depth maps estimated from higher quality LFs shall be closer to those estimated from ground-truth ones. Thus, we further compared the depth maps estimated from reconstructed LFs by different methods via an identical LF depth estimation method [63]. As shown in Fig. 8, our method can produce depth maps with sharper edges at occlusion boundaries and preserve smoothness at regions with uniform depth, which are closest to the ground truth ones. Such observations also demonstrate the advantage of our method on preserving the LF parallax structure. Besides, as the ground-truth depth maps of the testing LF images are available, we also quantitatively compared the accuracy of depth estimated from reconstructed LFs by different methods. To be specific, we calculated the Mean Square error (MSE) between the estimated depth maps and ground-truth ones. As shown in Table II, our method produces lower MSE values than all of the compared methods.

5) Efficiency Comparison: We compared the efficiency and model size of different methods. We implemented all the methods on a Linux server with Intel CPU E5-2699 @ 2.20GHz, 128GB RAM and Tesla V100. As listed in Table III, we can see that our method is much faster than Wu et al. [34] but slower than the other methods. Besides, our model size is smaller than Bao et al. [57], but larger than the other methods. Although our method is able to reconstruct novel views with much higher quality, there is still room to improve its efficiency.

C. Evaluation on Multi-View Datasets

We evaluated our method on two multi-view datasets, i.e., DTU [62] and RealEstate10K [11]. The transformation between views in these datasets has more degrees of freedom than that of LF datasets, leading to more complicated parallax structure, and thus more challenging for view synthesis methods. On the DTU dataset [62], we compared the proposed method with three IBR methods named FVS [16], SVNVS [17], and Guo et al. [18], as well as two NeRF-based methods named pixelNeRF [44] and IBRNet [13]. On the RealEstate10K dataset [11], we compared the proposed method with SVNVS [17], Guo et al. [18], and IBRNet [13]. We did not compare the proposed method with FVS [16] and pixelNeRF [44] on the RealEstate10K dataset [11] since they require either off-the-shelf depth maps or scale matrix which are not provided by the RealEstate10K dataset [11]. For fair comparisons, we trained the compared methods on the same dataset as ours with the officially released codes and suggested configurations.

1) Quantitative Comparison: We quantitatively compared different methods in terms of PSNR and SSIM in Tables IV and V, where it can be observed that

- our method achieves higher average PSNR and SSIM values than our preliminary conference version Guo et al.

TABLE I
QUANTITATIVE COMPARISONS (PSNR/SSIM) OF DIFFERENT METHODS ON THE INRIA SPARSE LF DATASET [55]. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

| Light Field          | Disparity range | Baseline (Warp) | Baseline (Disparity) | Kalantari et al. [33] | Wu et al. [34] | Wu et al. [36] | Jin et al. [36] | Guo et al. [18] | Bao et al. [57] | Ours   |
|----------------------|-----------------|-----------------|----------------------|-----------------------|---------------|---------------|---------------|----------------|----------------|--------|
| Electro_devices      | [19.4, 32.8]    | 32.87/0.943     | 32.99/0.941          | 24.58/0.691           | 29.12/0.865    | 31.22/0.883   | 33.04/0.935   | 35.38/0.959    | 32.81/0.929    | 36.25/0.963 |
| Flying_furniture     | [34.0, 62.4]    | 31.04/0.901     | 32.24/0.896          | 28.99/0.786           | 27.31/0.795    | 31.19/0.858   | 31.35/0.892   | 33.64/0.932    | 31.51/0.892    | 35.75/0.947 |
| Coffee_beans_vases   | [10.8, 28.4]    | 27.43/0.920     | 29.13/0.934          | 21.56/0.582           | 25.42/0.890    | 26.57/0.856   | 28.22/0.928   | 29.37/0.942    | 29.55/0.932    | 30.45/0.950 |
| Dinosaur             | [57.6, 72.8]    | 25.45/0.865     | 27.20/0.896          | 22.40/0.730           | 21.81/0.784    | 24.14/0.881   | 27.23/0.900   | 29.19/0.902    | 26.71/0.889    | 28.41/0.924 |
| Flowers              | [40.4, 66.6]    | 23.86/0.806     | 24.41/0.822          | 22.02/0.669           | 23.45/0.779    | 23.73/0.817   | 24.35/0.839   | 24.80/0.845    | 26.74/0.878    | 25.13/0.868 |
| Rooster_clock        | [34.4, 21.2]    | 30.46/0.952     | 35.77/0.953          | 27.73/0.710           | 29.41/0.889    | 25.15/0.884   | 27.58/0.928   | 38.48/0.966    | 35.38/0.942    | 38.26/0.968 |
| Smiling_crowd        | [40.4, 64.8]    | 20.75/0.822     | 21.29/0.818          | 17.01/0.602           | 20.33/0.756    | 20.36/0.777   | 21.01/0.619   | 22.34/0.868    | 20.86/0.816    | 22.79/0.877 |
| **Average**          |                 | 28.27/0.887     | 29.00/0.894          | 22.76/0.682           | 25.61/0.827    | 26.34/0.851   | 25.74/0.891   | 30.17/0.916    | 29.08/0.897    | 31.00/0.928 |

Fig. 5. Quantitative comparisons (PSNR/SSIM) of different methods under various disparity ranges (pixels) between input SAIs on the MPI LF dataset [61]. (a) Baseline (Warp), (b) Baseline (Disparity), (c) Kalantari et al. [33], (d) Wu et al. [34], (e) Jin et al. [36], (f) Guo et al. [18], (g) Bao et al. [57], (h) Ours. The two subfigures share the same legend.
Fig. 6. Visual comparisons of reconstructed SAIs from different methods on the Inria Sparse LF dataset [55]. (a) Ground Truth, (b) Baseline (Warp), (c) Baseline (Disparity), (d) Kalantari et al. [33], (e) Wu et al. [34], (f) Wu et al. [56], (g) Jin et al. [36], (h) Guo et al. [18], (i) Bao et al. [57], and (j) Ours. The disparity range between input SAIs of each LF is shown on the left.

| TABLE II | QUANTITATIVE COMPARISONS OF THE DEPTH MAPS ESTIMATED FROM THE GROUND-TRUTH LFS AND THE RECONSTRUCTED LFS BY DIFFERENT METHODS ON THE INRIA SPARSE LF DATASET [55]. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY |
|-----------------------------------------------|
| Baseline (Warp) | Baseline (Disparity) | Kalantari et al. [33] | Wu et al. [34] | Wu et al. [56] | Jin et al. [36] | Guo et al. [18] | Bao et al. [57] | Ours | GT |
|-----------------|----------------------|----------------------|----------------|----------------|----------------|----------------|----------------|------|-----|
| MSE             | 19.23                | 18.98                | 23.39          | 24.92          | 22.29          | 21.28          | 18.15          | 19.06 | 18.06 | 16.96 |

| TABLE III | COMPARISONS OF RUNNING TIME (IN SECONDS PER VIEW) AND MODEL PARAMETER SIZE (M) OF DIFFERENT METHODS ON THE INRIA SPARSE LF DATASET [55] |
|------------|---------------------------------------------------------------|
| Baseline (Warp) | Baseline (Disparity) | Kalantari et al. [33] | Wu et al. [34] | Wu et al. [56] | Jin et al. [36] | Guo et al. [18] | Bao et al. [57] | Ours | # Params |
| Time        | 7.64                                                           | 0.82                                                            | 6.32            | 27.77          | 2.82            | 0.69            | 2.78            | 0.21 | 11.01 |
| # Params    | 7.25                                                           | 2.52                                                            | 2.55            | 0.24           | 0.55            | 2.22            | 0.69            | 24.03| 7.45  |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 7. Visual comparisons of reconstructed SAIs from different methods on the MPI LF dataset [61]. (a) Ground Truth, (b) Baseline (Warp), (c) Baseline (Disparity), (d) Kalantari et al. [33], (e) Wu et al. [34], (f) Wu et al. [56], (g) Jin et al. [36], (h) Guo et al. [18], (i) Bao et al. [57], (j) Ours. The disparity range between input SAIs reaches 80 pixels for each reconstructed LF.

TABLE IV
QUANTITATIVE COMPARISONS (PSNR/SSIM) OF DIFFERENT METHODS ON THE DTU DATASET [62]. THE BEST AND SECOND BEST RESULTS ARE HIGHLIGHTED IN RED AND BLUE, RESPECTIVELY

| Scenes | FVS [16] | pixelNeRF [44] | IBRNet [13] | SVNVS [17] | Guo et al. [18] | Ours |
|--------|----------|----------------|-------------|-------------|-----------------|------|
| scan3  | 16.52/0.602 | 15.05/0.421 | 18.50/0.606 | 20.01/0.729 | 18.80/0.621 | 20.49/0.707 |
| scan5  | 16.30/0.575 | 16.30/0.562 | 20.65/0.658 | 20.75/0.752 | 19.13/0.632 | 21.40/0.733 |
| scan17 | 16.50/0.552 | 14.53/0.450 | 17.81/0.583 | 18.81/0.686 | 17.27/0.532 | 19.57/0.669 |
| scan21 | 15.28/0.525 | 13.43/0.404 | 15.88/0.520 | 16.91/0.654 | 15.07/0.492 | 16.73/0.603 |
| scan28 | 17.20/0.603 | 14.77/0.519 | 17.64/0.635 | 19.18/0.744 | 15.10/0.528 | 18.14/0.692 |
| scan35 | 21.09/0.723 | 13.66/0.571 | 21.26/0.733 | 19.51/0.766 | 20.42/0.714 | 22.91/0.787 |
| scan37 | 18.57/0.670 | 14.89/0.647 | 19.52/0.713 | 20.56/0.787 | 18.62/0.710 | 21.22/0.795 |
| scan38 | 18.73/0.591 | 15.92/0.554 | 20.74/0.634 | 21.80/0.749 | 20.11/0.612 | 22.52/0.730 |
| scan40 | 19.03/0.588 | 16.21/0.596 | 20.97/0.646 | 21.38/0.749 | 20.59/0.624 | 21.41/0.720 |
| scan43 | 17.34/0.671 | 16.22/0.568 | 19.83/0.705 | 20.57/0.778 | 18.85/0.676 | 21.21/0.786 |
| scan56 | 21.35/0.678 | 23.89/0.718 | 24.00/0.682 | 24.08/0.741 | 22.31/0.611 | 24.10/0.733 |
| scan59 | 18.06/0.699 | 16.90/0.660 | 20.32/0.769 | 21.84/0.832 | 17.53/0.685 | 22.35/0.837 |
| scan66 | 21.06/0.778 | 21.75/0.806 | 25.09/0.816 | 24.45/0.828 | 22.24/0.781 | 23.57/0.833 |
| scan67 | 19.45/0.751 | 20.33/0.802 | 25.00/0.807 | 22.83/0.808 | 21.47/0.767 | 22.79/0.813 |
| scan82 | 19.54/0.788 | 20.44/0.834 | 22.83/0.854 | 22.31/0.851 | 19.98/0.791 | 22.69/0.862 |
| scan86 | 26.29/0.767 | 26.62/0.797 | 28.58/0.797 | 29.78/0.822 | 28.35/0.776 | 29.95/0.816 |
| scan106 | 23.43/0.791 | 22.73/0.779 | 25.68/0.815 | 26.58/0.835 | 24.28/0.774 | 25.52/0.832 |
| scan117 | 22.73/0.776 | 22.87/0.780 | 26.69/0.818 | 26.47/0.829 | 25.16/0.780 | 26.71/0.833 |
| Average | 19.36/0.674 | 18.14/0.637 | 21.72/0.711 | 22.10/0.774 | 20.29/0.672 | 22.41/0.766 |
Fig. 8. Visual comparisons of estimated depth maps from ground-truth LFs and reconstructed LF by different methods on the Inria Sparse LF dataset [55]. (a) Ground Truth, (b) Baseline (Warp), (c) Baseline (Disparity), (d) Kalantari et al. [33], (e) Wu et al. [34], (f) Wu et al. [56], (g) Jin et al. [36], (h) Guo et al. [18], (i) Bao et al. [57], (j) Ours. The disparity range between input SAIs of each LF is shown on the left.

2) Comparisons of Visual Results: We compared the visual results of different methods on both the DTU dataset [62] and the RealEstate10K dataset [11] in Figs. 9 and 10, which further demonstrate the advantage of our proposed method. As can be seen, our method can synthesize high-quality details and structures at most areas, while the compared methods either produce severe distortions or blurry effects at these regions. Besides, our method can preserve sharp edges at occlusion boundaries, while other methods show severe ghost effects. We refer the readers to the Supplementary Material, available online, for more visual results.

3) Comparisons of Efficiency: We compared the efficiency and model size of different methods. We implemented all methods on a Linux server with Intel CPU E5-2620 @ 2.10GHz, 256GB RAM and GeForce RTX 2080 Ti. As listed in Table VI, our method is faster than pixelNeRF [44] while slower than the other compared methods, and our model size is larger than Guo et al. [18] while smaller than other methods. Taking the reconstruction quality, efficiency, and model size together, we believe our method is the best.

4) Comparisons of Different Numbers of Input Views: We evaluated the proposed method with 2, 3 and 4 source views fed
Fig. 9. Visual comparisons of synthesized views from different methods on the DTU dataset [62]. (a) Ground Truth, (b) FVS [16], (c) pixelNeRF [44], (d) IBRNet [13], (e) SVNVS [17], (f) Guo et al. [18], and (g) Ours.
on the DTU dataset [62]. Specifically, in addition to selecting two adjacent views of the target view as source views, we separately added one and two views as the additional source views for the 3-inputs and 4-inputs tasks, respectively, by using the view selection result provided by [4]. As listed in Table VII, it can be observed that our method can achieve better reconstruction quality along with the number of input views increasing. More specifically, the improvement is more obvious from 2-inputs to 3-inputs than that of from 3-inputs to 4-inputs. The reason may be that 3 inputs provide much more useful information than 2 inputs and is sufficient for the reconstructions of most target views. Adding another source view based on 3 inputs, i.e., 4 inputs, could not provide much useful information.

D. Ablation Study

To validate the effectiveness of the key components of our method, i.e., content embedding, global embedding, adaptive PSV fusion, feature-space warping, and weight-smoothness loss, we carried out comprehensive ablation studies on the DTU dataset [62] under the setting of 2 input views. To be specific, we sequentially added the modules to the base model one by one until all the five components were included to form the complete model.

Content Embedding. By comparing the results of Models #1 and #2 in Table VIII, it can be seen that there is a significant

---

**TABLE V**

| Scenes | IBRNet [13] | SVNVS [17] | Guo et al. [18] | Ours |
|--------|-------------|-------------|-----------------|------|
| RE10K-1 | 35.58/0.976 | 32.77/0.959 | 39.76/0.986 | 40.09/0.987 |
| RE10K-2 | 28.04/0.954 | 22.42/0.873 | 31.91/0.971 | 31.94/0.970 |
| RE10K-3 | 33.44/0.994 | 28.76/0.948 | 46.96/0.966 | 45.31/0.995 |
| RE10K-4 | 27.11/0.915 | 24.14/0.852 | 29.32/0.947 | 29.81/0.955 |
| RE10K-5 | 29.71/0.892 | 28.73/0.919 | 33.14/0.958 | 36.12/0.966 |
| RE10K-6 | 33.36/0.985 | 32.06/0.899 | 38.48/0.952 | 39.03/0.994 |
| RE10K-7 | 27.61/0.836 | 26.48/0.861 | 31.93/0.930 | 31.11/0.915 |
| RE10K-8 | 30.19/0.917 | 24.51/0.914 | 33.21/0.963 | 34.66/0.960 |
| RE10K-9 | 28.65/0.929 | 28.98/0.920 | 31.74/0.957 | 32.00/0.961 |
| RE10K-10 | 23.28/0.772 | 22.99/0.820 | 27.55/0.895 | 27.33/0.898 |
| RE10K-11 | 24.26/0.882 | 22.88/0.849 | 25.45/0.888 | 25.33/0.891 |
| RE10K-12 | 27.06/0.916 | 30.66/0.955 | 36.10/0.952 | 36.94/0.986 |
| RE10K-13 | 31.47/0.947 | 32.34/0.960 | 38.48/0.964 | 38.50/0.984 |
| RE10K-14 | 25.05/0.851 | 29.52/0.943 | 33.76/0.972 | 35.10/0.979 |
| RE10K-15 | 26.87/0.862 | 24.55/0.886 | 29.88/0.939 | 30.82/0.949 |
| RE10K-16 | 35.10/0.971 | 34.36/0.957 | 41.48/0.984 | 42.03/0.986 |
| RE10K-17 | 35.87/0.975 | 31.34/0.960 | 38.63/0.964 | 39.97/0.967 |
| Average | 29.70/0.914 | 28.09/0.914 | 34.81/0.960 | 33.09/0.962 |

**TABLE VI**

| FVS [16] | pixelNeRF [44] | IBRNet [13] | SVNVS [17] | Guo et al. [18] | Ours |
|----------|----------------|-------------|-------------|-----------------|------|
| Time     | 0.14           | 14.29       | 2.82        | 0.15            | 1.02 |
| # Params | 33.73          | 28.16       | 8.96        | 153.15          | 0.69 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
TABLE VIII
QUANTITATIVE RESULTS OF THE ABALATION STUDIES ON THE FIVE KEY COMPONENTS OF OUR METHOD

| Model number | Content embedding | Global embedding | PSV fusion | Feature-space warping | Weight-smoothness loss | PSNR/SSIM |
|--------------|-------------------|------------------|------------|-----------------------|------------------------|-----------|
| 1            | x                 | x                | x          | x                     | x                      | 20.69/0.704 |
| 2            | ✔                 | x                | x          | x                     | x                      | 21.85/0.750 |
| 3            | ✔                 | ✔                | x          | x                     | x                      | 21.94/0.751 |
| 4            | ✔                 | ✔                | ✔          | x                     | x                      | 22.15/0.759 |
| 5            | ✔                 | ✔                | ✔          | ✔                     | x                      | 22.30/0.767 |
| 6            | ✔                 | ✔                | ✔          | ✔                     | ✔                      | 22.41/0.766 |

“x” Denotes that the corresponding component is not included, while “✔” denotes being included.

Fig. 11. Visual illustration of the ablation studies on the five key components of our framework.

increase of performance when adding the content embedding to the base model, which verifies the advantage brought by detecting the texture edges of input views, and understanding the occlusion and non-Lambertian relations between input views. Fig. 11(a) and (b) also visually verify the advantage.

Global Embedding. By comparing the results of Models #2 and #3 in Table VIII, this component involved in the spatial refinement module can improve the synthesis performance. As shown in Fig. 11(c) and (d), we can see that the results of the framework with global embedding are closer to the ground truth at the occlusion boundary and reflecting area.

PSV Fusion. According to the results of Models #3 and #4 in Table VIII, this component involved in the spatial refinement module can improve the PSNR value by about 0.19 dB. As shown in Fig. 11(e) and (f), the results without this component are obviously blurry at the occlusion boundaries and the texture regions.

Feature-Space Warping. We can validate the advantage of this component contained in the spatial refinement module by comparing the results of Models #4 and #5 in Table VIII. As shown in Fig. 11(g) and (h), without this component, some fine structures, such as delicate objects and textures, are obviously broken.

Weight-Smoothness Loss. The effectiveness of this loss term is validated by comparing the results of Models #5 and #6 in Table VIII. Besides, as shown in Fig. 11(i) and (j), adding this component generates more visually-pleasing results than being without it, which also verifies the advantage.

E. Limitation

According to Table IV, our method is inferior to IBRNet [13] and SVNVS [17] on some scenes of the DTU dataset, e.g., scan66, scan67 and scan106 which contain many texture-less or repeated texture regions. For example, as shown in Fig. 12, it can be observed that our method achieves better visual results than FVS [16] and pixelNeRF [44], but worse than IBRNet [13].
and SVNVS [17]. The possible reason is that our content-aware warping performs the interpolation only in a local neighborhood and predicts interpolation weights for each neighboring pixel only conditioned on its local information, and thus, the predicted weights are evenly distributed to all the neighboring pixels with very close intensities at these regions, which results in blurring effects.

Besides, the quality of synthesized views by our method on the scenes with too various relative camera pose patterns between the source view and the target view is still limited. To illustrate this limitation, we retrained and tested our network on the Tanks and Temples dataset [64] under the setting of 2 input views. Note that unlike the DTU [62] and RealEstate10K [11] datasets, where the distributions of cameras poses are same on all scenes, or the camera is always moving forward, the Tanks and Temples dataset [64] has more complex and irregular camera trajectories, resulting in more various relative camera pose patterns between the source view and the target view, making it more challenging to synthesize high-quality views. Specifically, following the method FVS [16], we split the Tanks and Temples dataset [64] into training and testing datasets, i.e., 17 out of 21 scenes for training and the remaining 4 scenes for testing. In the training phase, we adopted the selection strategy provided by FVS [16], which counts the number of pixels from the target view that are mapped to the valid source image domain.

We have presented an end-to-end learning-based framework for novel view synthesis from a set of input source views. Particularly, we proposed learnable content-aware warping to overcome the natural limitations of the traditional image warping operation. Owing to the content-aware warping, as well as the blending and refinement modules that are elaborately designed to handle occlusions and recover spatial correlations, the proposed view synthesis framework reconstructs novel views with much higher quality on both LF datasets and multi-view datasets, compared with state-of-the-art methods.

VI. CONCLUSION

We have presented an end-to-end learning-based framework for novel view synthesis from a set of input source views. Particularly, we proposed learnable content-aware warping to overcome the natural limitations of the traditional image warping operation. Owing to the content-aware warping, as well as the blending and refinement modules that are elaborately designed to handle occlusions and recover spatial correlations, the proposed view synthesis framework reconstructs novel views with much higher quality on both LF datasets and multi-view datasets, compared with state-of-the-art methods.

REFERENCES

[1] R. Szeliski, Computer Vision: Algorithms and Applications. Berlin, Germany: Springer, 2010.
[2] Y. Furukawa and J. Ponce, “Accurate, dense, and robust multiview stereopsis,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 8, pp. 1362–1376, Aug. 2010.
[3] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 4104–4113.
[4] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, “MVSNet: Depth inference for unstructured multi-view stereo,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 767–783.
[5] C. Guo, J. Jin, J. Hou, and J. Chen, “Accurate light field depth estimation via an occlusion-aware network,” in Proc. IEEE Int. Conf. Multimedia Expo, 2020, pp. 1–6.
[6] F.-C. Huang, K. Chen, and G. Wetzstein, “The light field stereoscope: Immersive computer graphics via factored near-eye light field displays with focus cues,” ACM Trans. Graph., vol. 34, no. 4, 2015, Art. no. 60.
[7] J. Yu, “A light-field journey to virtual reality,” IEEE MultiMedia, vol. 24, no. 2, pp. 104–112, Second Quarter 2017.
[8] S.-E. Wei et al., “VR facial animation via multiview image translation,” *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–16, 2019.

[9] P. E. Debevec, C. J. Taylor, and J. Malik, “Modeling and rendering architecture from photographs: A hybrid geometry-and-image-based approach,” in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 11–20.

[10] M. Levoy and P. Hanrahan, “Light field rendering,” in *Proc. 23rd Annu. Conf. Comput. Graph. Interactive Techn.*, 1996, pp. 31–42.

[11] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely, “Stereo magnification: Learning view synthesis using multplane images,” *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, 2016.

[12] B. Mildenhall et al., “Local light field fusion: Practical view synthesis with prescriptive sampling guidelines,” *ACM Trans. Graph.*, vol. 38, no. 4, pp. 1–14, 2019.

[13] Q. Wang et al., “IBRNet: Learning multi-view image-based rendering,” in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 8, Jul. 2023, pp. 3261–3273, Jul. 2019.

[14] P. P. Srinivasan, T. Wang, A. Sreelal, R. Ramamoorthi, and N. Ng, “Learning to synthesize a 4D RGBD light field from a single image,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2243–2251.

[15] J. Jin, J. Hou, J. Chen, H. Zeng, S. Kwong, and J. Yu, “Deep coarse-to-fine dense light field reconstruction with flexible sampling and geometry-aware fusion,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 1819–1836, Apr. 2022.

[16] G. Penner and L. Shi, “Soft 3D reconstruction for view synthesis,” *ACM Trans. Graph.*, vol. 36, no. 6, pp. 1–11, 2017.

[17] P. Hedman, J. Philip, T. Price, J.-M. Frahm, G. Drettakis, and G. Brostow, “Deep blending for free-viewpoint image-based rendering,” *ACM Trans. Graph.*, vol. 37, no. 6, pp. 1–15, 2018.

[18] I. Choi, O. Gallo, A. Troccoli, M. H. Kim, and J. Kautz, “Extreme view synthesis,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 7781–7790.

[19] J. Flynn et al., “DeepView: View synthesis with learned gradient descent,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2367–2376.

[20] P. P. Srinivasan, R. Tucker, J. T. Barron, R. Ramamoorthi, R. Ng, and N. Snavely, “Pushing the boundaries of view extrapolation with multplane images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 175–184.

[21] R. Tucker and N. Snavely, “Single-view view synthesis with multplane images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 551–560.

[22] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, “NeRF: Representing scenes as neural radiance fields for view synthesis,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 405–421.

[23] A. Yu, V. Ye, M. Tancik, and A. Kanazawa, “pixelNeRF: Neural radiance fields from one or few images,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4578–4587.

[24] T. Porter and T. Duff, “Compositing digital images,” in *Proc. 11th Annu. Conf. Comput. Graph. Interactive Techn.*, 1984, pp. 253–259.

[25] J. T. Kajiya and B. P. Von Herzen, “Ray tracing volume densities,” *ACM SIGGRAPH Comput. Graph.*, vol. 18, no. 3, pp. 165–174, 1984.

[26] M. Jaderberg et al., “Spatial transformer networks,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[27] S. Bako et al., “Kernel-predicting convolutional networks for denoising Monte Carlo renderings,” *ACM Trans. Graph.*, vol. 36, no. 4, 2017, Art. no. 97.

[28] Z. Teed and J. Deng, “RAFT: Recurrent All-Pairs field transforms for optical flow,” in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2020, pp. 402–419.

[29] S.-H. Sun, M. Huh, Y.-H. Liao, N. Zhang, and J. J. Lim, “Multi-view to novel view: Synthesizing novel views with self-learned confidence,” in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 155–171.

[30] R. T. Collins, “A space-sweep approach to true multi-image matching,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1996, pp. 358–363.

[31] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–9.

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[33] M. Guo, J. Hou, J. Jin, J. Chen, and L.-P. Chau, “Deep spatial-angular regularization for light field imaging, denoising, and super-resolution,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6094–6110, Oct. 2022.

[34] J. Shi, X. Jiang, and C. Guillemot, “A framework for learning depth from a flexible subset of dense and sparse light field views,” *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 5867–5880, Dec. 2019.

[35] G. Wu, Y. Liu, L. Fang, and T. Chai, “Revisiting light field rendering with deep anti-aliasing neural network,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 9, pp. 5430–5444, Sep. 2022.

[36] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, “Depth-aware video frame interpolation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3703–3712.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[38] G. Riegler and V. Koltun, “Stable view synthesis,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12 216–12 225.
Mantang Guo received the BEng and MEng degrees from the School of Computer Science, Northwestern Polytechnical University, China, in 2016 and 2019, respectively. He is currently working toward the PhD degree with the Department of Computer Science, City University of Hong Kong. His research interests include computational photography, light field reconstruction, and deep learning.

Junhui Hou (Senior Member, IEEE) received the BEng degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the MEng degree in signal and information processing from Northwestern Polytechnical University, Xi’an, China, in 2012, and the PhD degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2016. He immediately joined the Department of Computer Science, City University of Hong Kong, as an assistant professor in January 2017. His research interests fall into the general areas of multimedia signal processing, such as image/video/3D geometry data representation, processing and analysis, semi/un-supervised data modeling, and data compression. He was the recipient of several prestigious awards, including the Chinese Government Award for Outstanding Students Study Abroad from China Scholarship Council in 2015 and the Early Career Award (3/381) from the Hong Kong Research Grants Council in 2018. He is an elected member of MSA-TC and VSPC-TC, and MMSP-TC. He is currently an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, Signal Processing: Image Communication, and The Visual Computer. He also served as the guest editor of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing and an area chair of ACM MM’19/20/21, IEEE ICME’20, VCIP’20/21, and WACV’21.

Jing Jin received the BEng degree from the Southeast University, Nanjing, China, in 2017, and the PhD degree from the Department of Computer Science, City University of Hong Kong, Hong Kong SAR, in 2021. Her research interests include light field image representation and processing.

Hui Liu received the BSc degree in communication engineering from the Central South University, Changsha, China, the MEng degree in computer science from Nanyang Technological University, Singapore, and the PhD degree from the Department of Computer Science, City University of Hong Kong, Hong Kong. She is currently a lecturer with the School of Computing Information Sciences, Caritas Institute of Higher Education, Hong Kong. From 2014 to 2017, she was a research associate with the Maritime Institute of Nanyang Technological University. Her research interests include image processing and machine learning.

Huanqiang Zeng (Senior Member, IEEE) received the BS and MS degrees from Huaqiao University, Xiamen, China, and the PhD degree from Nanyang Technological University, Singapore, all in electrical engineering. He is currently a full professor with the School of Information Science and Engineering, Huaqiao University, Xiamen, China. Before that, he was a postdoctoral fellow with the Chinese University of Hong Kong, Hong Kong. His research interests include image processing, video coding, machine learning, and computer vision. He has been actively serving as the associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, and IET Electronics Letters.

Jiwen Lu (Senior Member, IEEE) is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He serves as the general co-chair for the International Conference on Multimedia and Expo (ICME) 2022, the program co-chair for the International Conference on Multimedia and Expo 2020, the International Conference on Automatic Face and Gesture Recognition (FG) 2023, and the International Conference on Visual Communication and Image Processing (VCIP) 2022. He serves as the co-editor-in-chief of the Pattern Recognition Letters, an associate editor of IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, and IEEE Transactions on Biometrics, Behavior, and Identity Sciences, and Pattern Recognition. He was a recipient of the National Natural Science Funds for Distinguished Young Scholar. He is a fellow of IAPR.