Light Field Super-Resolution via Adaptive Feature Remixing

Keunsoo Ko, *Student Member, IEEE,* Yeong Jun Koh, *Member, IEEE,* Soonkeun Chang, *Member, IEEE,* and Chang-Su Kim, *Senior Member, IEEE*

***Abstract*—A novel light field super-resolution algorithm to improve the spatial and angular resolutions of light field images is proposed in this work. We develop spatial and angular super-resolution (SR) networks, which can faithfully interpolate images in the spatial and angular domains regardless of the angular coordinates. For each input image, we feed adjacent images into the SR networks to extract multi-view features using a trainable disparity estimator. We concatenate the multi-view features and remix them through the proposed adaptive feature remixing (AFR) module, which performs channel-wise pooling. Finally, the remixed feature is used to augment the spatial or angular resolution. Experimental results demonstrate that the proposed algorithm outperforms the state-of-the-art algorithms on various light field datasets. The source codes and pre-trained models are available at https://github.com/keunsoo-ko/LFSR-AFR**

***Index Terms*—Light field, super-resolution, feature remixing, convolutional neural network (CNN).**

I. Introduction

A LIGHT field (LF) records the intensity and direction of light rays, which are reflected from objects in 3D environments. Unlike the conventional imaging that records the 2D projection of light rays, LF imaging captures high dimensional data [1]. From the high dimensional data, we can extract spatial and angular information of light rays, and thus can reconstruct multi-view images of a scene. This rich visual information in LF images can facilitate many image processing and computer vision tasks [2]–[5].

However, acquiring LF data with plenoptic cameras, such as Lytro [6] and Raytrix [7], suffers from the trade-off between spatial and angular resolutions. Due to a limited sensor resolution, a plenoptic camera should lower the spatial resolution of each view to capture more views with a higher angular sampling rate, or vice versa. Low-resolution images lead to performance degradation of LF vision applications. It is hence necessary to enhance the resolutions of LF images. This paper addresses the problem of LF super-resolution (LFSR).

Multi-view images in LF are highly correlated to one another. Hence, sub-pixel information in each view image can be estimated by exploiting this cross-view correlation, thereby enabling its super-resolution (SR) reconstruction. LFSR algorithms predict sub-pixel information using the disparity between neighboring views [8]–[11]. Recently, many deep learning algorithms with different network architectures [12]–[17] have been developed to achieve LFSR using large LF datasets [18]–[20]. These algorithms yield reliable SR results by utilizing the cross-view correlation through convolutional neural networks (CNNs). However, since the number of adjacent images, required to super-resolve a view, varies according to the angular coordinates of the view, some algorithms [12], [16] should train several networks separately.

In this paper, we propose two networks to achieve spatial and angular SR based on adaptive feature remixing (AFR), which yield high quality super-resolved images regardless of the angular coordinates of input view images. The proposed spatial and angular SR networks take multi-view images to enhance the spatial resolution and expand the angular resolution, respectively. We first extract disparity-compensated multi-view features using a trainable disparity estimator and concatenate the multi-view features. Next, we use the proposed AFR module to perform channel-wise pooling and remix the concatenated feature according to the angular coordinates of the input view image. Finally, the remixed feature is used to yield a super-resolved image. More specifically, when enhancing the spatial resolution, an up-sample scheme generates the high-resolution image using the remixed feature. On the other hand, when augmenting the number of views in the angular domain, the remixed feature is adopted to produce blending filters. Then, reference images are superposed by the blending filters to reconstruct augmented view images. Experimental results demonstrate that the proposed spatial and angular SR networks outperform the state-of-the-art algorithms on various LF datasets [18]–[21].

To summarize, this work has three main contributions.

- We develop the spatio-angular SR algorithm that improves the spatial and angular resolutions of low-resolution LF images.
- We propose the AFR scheme, which enables to super-resolve input views regardless of their angular coordinates using a single network.
- The proposed algorithm provides remarkable performances of spatial and angular SR on the LF datasets in [18]–[21].

II. Related Work

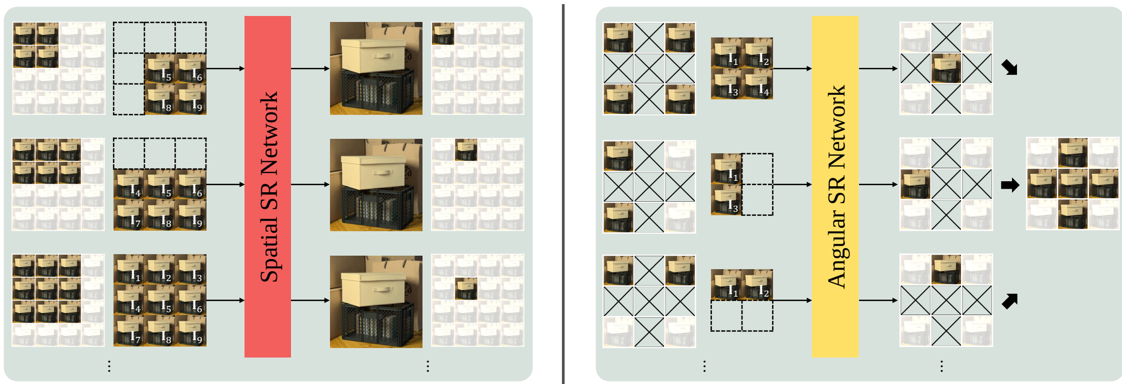
Single-image SR: Extensive researches have been carried out to perform single-image SR, including elementary interpola-

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP)(No. NRF-2018R1A2B3003896 and No. NRF-2019R1F1A1062907).

K. Ko and C.-S. Kim are with the School of Electrical Engineering, Korea University, Seoul, Korea. (e-mail: ksko@mcl.korea.ac.kr; changsukim@korea.ac.kr)

Y. J. Koh is with the Department of Computer Science & Engineering, Chungnam National University, Daejeon, Korea. (e-mail: yjkoh@cnu.ac.kr)

S. Chang is with Samsung Electronics Co., Ltd., Gyeonggi-do, Korea. (e-mail: sk107.chang@samsung.com)



(a) (b)

Fig. 1: Overview of the proposed algorithm. In (a), an LF consists of 4×4 view images. To increase the spatial resolution of each image $I_{\mathbf{u}}$, the spatial SR network takes the 3×3 images around $I_{\mathbf{u}}$ as input. In (b), the angular SR network processes 2×2 input images to reconstruct 5 intermediate images, resulting in 3×3 images. For both SR, some adjacent views may be unavailable. In such cases, virtual images are used instead, whose all pixel values are zeros.

tion [22], [23], self-similarity [24], [25] and dictionary learning [26], [27] methods. Motivated by the success of CNNs, Dong *et al.* [28] first introduced a CNN-based SR algorithm using a shallow network. Many deeper CNN structures have been proposed to improve the SR performance [29]–[35].

LFSR: The objective of LFSR is to improve the resolutions of multi-view images, recorded at low-resolutions. To restore sub-pixel information in multi-view images, disparity vectors between neighboring view images are estimated [8]–[11]. Bishop and Favaro [8] reconstructed a depth map from disparity vectors and used the depth information to estimate a space-varying point spread function for SR. Mitra and Veeraraghavan [9] designed Gaussian mixture models (GMMs) for LF patches using disparity vectors and reconstructed high-resolution patches based on the GMMs. Wanner and Goldluecke [10] obtained dense disparity maps for LFSR, by analyzing the structure tensor of epipolar plane images. Rossi and Frossard [11] proposed a global optimization method, which forms a warping matrix based on coarse disparities, to achieve LFSR.

There is also the data-driven approach that learns the mapping between low and high-resolution LF images. Farrugia *et al.* [36] learned a subspace to obtain high-resolution images based on multivariate ridge regression. Yoon *et al.* [12] presented an early deep learning scheme for LFSR. Fan *et al.* [13] applied the single-image SR algorithm in [29] to each view image separately and then improved the qualities of the separate high-resolution images using the multi-patch fusion CNN. Gul and Gunturk [14] used raw LF data directly as the input to CNNs to enhance the spatial and angular resolutions. Wang *et al.* [15] developed the bidirectional recurrent CNN to generate horizontally and vertically up-sampled image stacks and combined them via stacked generalization. Zhang *et al.* [16] stacked CNN features of multiple view images to exploit residual information between neighboring views and to generate LFSR results. Yeung *et al.* [37] proposed the spatial-angular separable convolution layer to process all views of an LF simultaneously. They improved the processing speed by approximating the 4D convolution layer with 2D convolution layers. Farrugia *et al.* [17] employed optical flow to align LF images and reduced the angular dimension using low-rank approximation. Then, they trained an embedding space using the low-rank model to reconstruct SR images.

III. Proposed Algorithm

We adopt the 4D LF representation in [38]. Specifically, we represent an LF by a four-dimensional three-channel signal

$$\mathbf{L}(u, v, x, y) \in \mathbb{R}^3, \tag{1}$$

which is defined on the domain $\mathbb{N}_U \times \mathbb{N}_V \times \mathbb{N}_W \times \mathbb{N}_H$ and yields color coordinates, such as RGB values, for each (u, v, x, y) in the domain. Here, $\mathbb{N}_k \triangleq \{1, 2, \ldots, k\}$. Also, (u, v) and (x, y) are angular and spatial coordinates, respectively. Thus, there are $U \times V$ view images and the spatial resolution of each image is $W \times H$. While fixing the angular resolution, we attempt to reconstruct a higher spatial resolution signal

$$\mathbf{L}_{\mathcal{S}}^{\mathrm{HR}}(u, v, x, y) \in \mathbb{R}^3, \tag{2}$$

defined on $\mathbb{N}_U \times \mathbb{N}_V \times \mathbb{N}_{r_s W} \times \mathbb{N}_{r_s H}$. Here, r_s is a scale factor for the spatial resolution. On the other hand, while fixing the spatial resolution, we try to reconstruct a higher angular resolution signal

$$\mathbf{L}_{\mathcal{A}}^{\mathrm{HR}}(u, v, x, y) \in \mathbb{R}^3, \tag{3}$$

defined on $\mathbb{N}_{r_a U} \times \mathbb{N}_{r_a V} \times \mathbb{N}_W \times \mathbb{N}_H$. Here, r_a is a scale factor for the angular resolution.

As done in [15], [16], [37], we convert RGB images into the YCbCr color space and focus on super-resolving Y images only. Cb and Cr images are simply up-sampled by the bicubic interpolation. Let $I_{\mathbf{u}}$ be the Y image of the $\mathbf{u}$-th view image in $\mathbf{L}$, where $\mathbf{u} \triangleq (u, v)$. Fig. 1(a) shows an overview of the proposed spatial SR network, which processes $\{I_{\mathbf{u}}\} \in \mathbf{L}$ to yield $\{I_{\mathbf{u}}^{\mathrm{HR}}\} \in \mathbf{L}_{\mathcal{S}}^{\mathrm{HR}}$. Also, Fig. 1(b) illustrates the proposed angular SR network, which super-resolves the angular resolution from 2×2 to 3×3 view images. Let us describe the proposed spatial and angular SR networks subsequently.

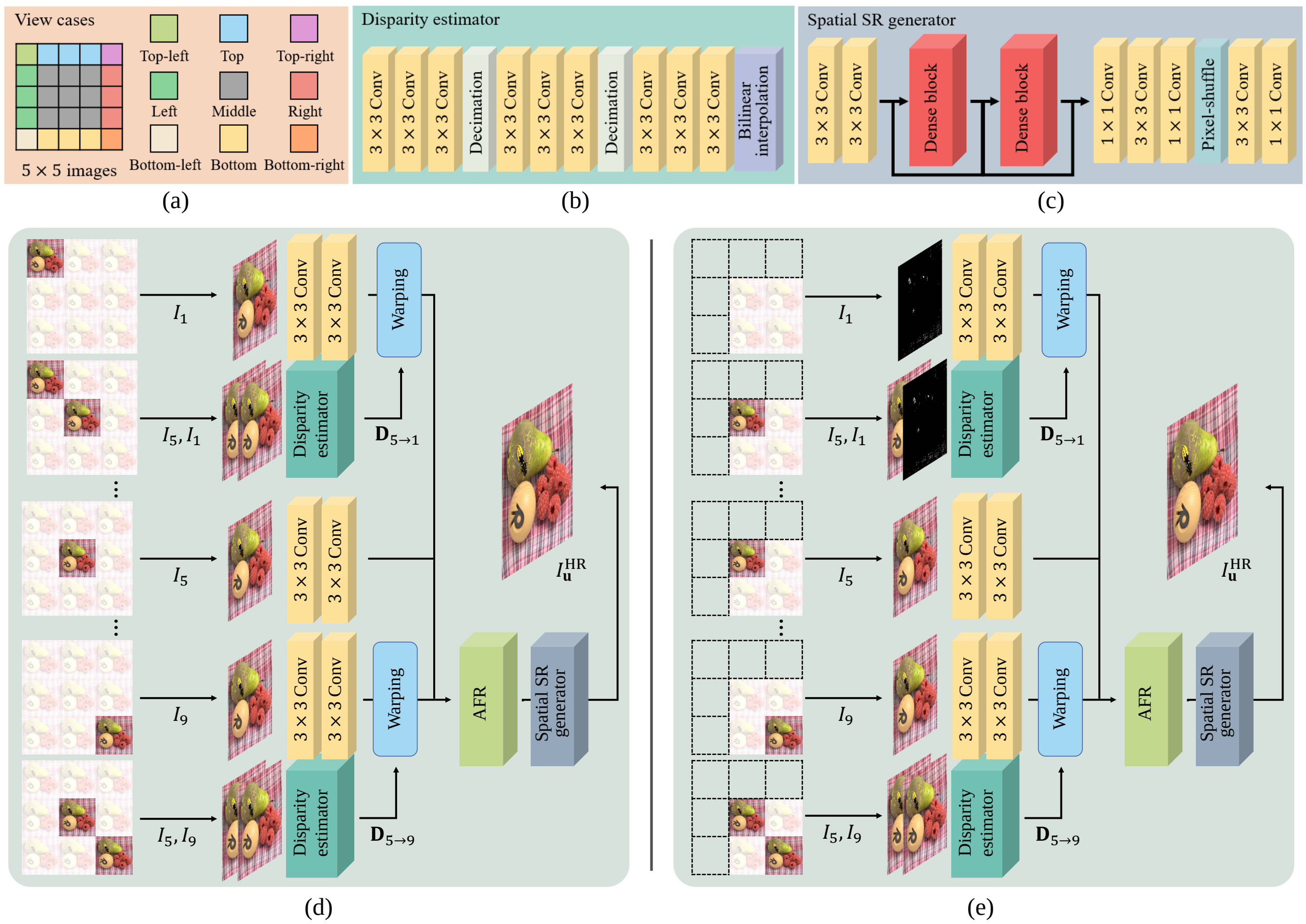


Fig. 2: (a) 5×5 view images, which are divided into nine cases, (b) disparity estimator, (c) spatial SR generator, (d) spatial SR for the middle case, and (e) spatial SR for the top-left case.

A. Spatial SR Network

Fig. 2 shows the architecture of the spatial SR network that performs three steps: multi-view feature extraction, AFR, and upsampling.

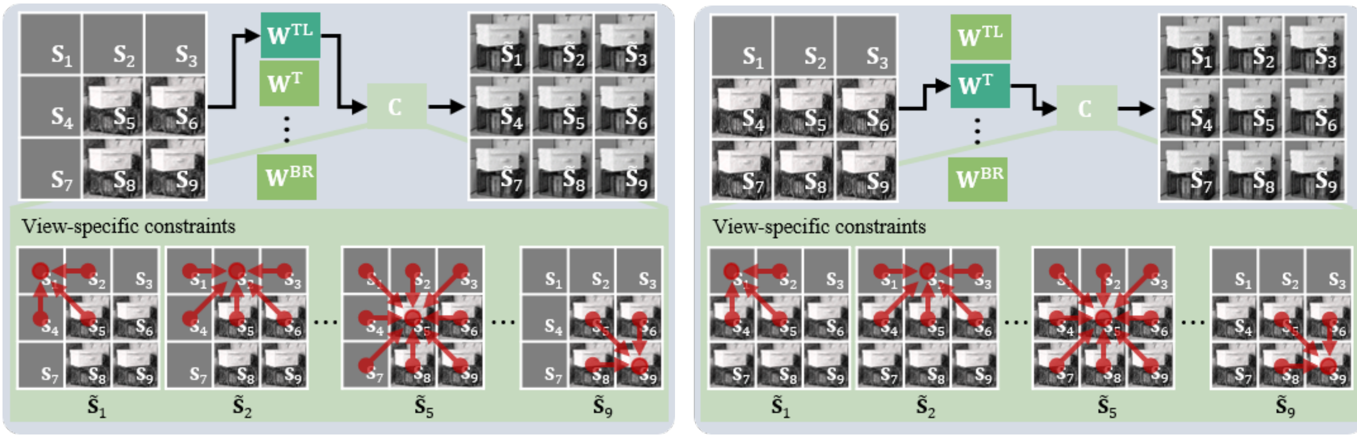
Multi-view feature extraction: We enhance the spatial resolution of each view image $I_{\mathbf{u}}$, $\mathbf{u} \in \mathbb{N}_U \times \mathbb{N}_V$, by exploiting the information in the 8-adjacent view images in the angular domain. For simpler notations, let $\mathcal{I}_{\mathbf{u}} = \{I_i\}_{i=1}^{9}$ denote the set of 3×3 images, composed of $I_{\mathbf{u}}$ and its 8-adjacent images. They are indexed from top-left to bottom-right, as illustrated in Fig. 1(a). Thus, $I_5 = I_{\mathbf{u}}$ is the image to be super-resolved. Also, for example, I_2 and I_4 are the top and left images of I_5, respectively. We also consider the cases that some adjacent images are unavailable. These cases occur when $\mathbf{u}$ is on the boundary of the angular domain $\mathbb{N}_U \times \mathbb{N}_V$. We fill in those missing images with virtual images, whose all pixel values are zeros.

The adjacent images in $\mathcal{I}_{\mathbf{u}}$ contain sub-pixel information for the central image I_5 with different offsets. Each adjacent image has different sub-pixel shifts according to its angular coordinates. For example, for I_5, the left and right images (I_4 and I_6) have sub-pixel information in the horizontal direction, while the top and bottom images (I_2 and I_8) do in the vertical one. Therefore, to exploit the sub-pixel information, we extract multi-view features by feeding all nine images to different network branches in Fig. 2(d). Then, we warp the extracted feature of each adjacent image I_i, $i \neq 5$, to match the central image I_5. For the warping, we estimate sub-pixel offsets between I_5 and I_i using a single disparity estimator in Fig. 2(b). Those sub-pixel offsets are called the disparities.

We design the disparity estimator in Fig. 2(b) with three successive convolution blocks, each of which has three convolution layers. The output of the last convolution block is up-sampled to be of the same size as the two images I_5 and I_i using bilinear interpolation. Let $\mathbf{F}_i$ and $\mathbf{D}_{5\to i}$ denote the extracted feature from I_i and the disparity map from I_5 to I_i, respectively. Then, $\mathbf{F}_i$ can be aligned spatially to the central view as follows:

$$\mathbf{S}_i = \mathcal{W}(\mathbf{F}_i, \mathbf{D}_{5\to i}) \tag{4}$$

where $\mathcal{W}$ is the backward warping function based on bilinear interpolation. Thus, $\mathbf{S}_i$ is the aligned feature of I_i. It has $C = 32$ channels with width W and height H. Then, we concatenate the spatially aligned multi-view features into $\mathbf{S} = \mathbf{S}_1 \parallel \mathbf{S}_2 \parallel \cdots \parallel \mathbf{S}_9$, where $\parallel$ denotes the concatenation



(a) (b)

Fig. 3: Examples of the AFR processes (a) for the top-left (TL) case and (b) for the top (T) case, respectively, in which different trainable matrices $\mathbf{W}^{\mathrm{TL}}$ and $\mathbf{W}^{\mathrm{T}}$ are used. Red arrows depict view-specific constraints.

along the channel dimension. Thus, $\mathbf{S}$ has $9C$ channels with the spatial resolution $W \times H$.

The proposed disparity estimator is trained within the spatial SR network in an end-to-end manner. Therefore, it estimates disparities that are tailored for SR, and the estimated disparities convey sub-pixel features to the central image effectively. To reduce the overall complexity, the disparity estimator is designed to have a much simpler structure than the conventional optical flow networks [39], [40]. Also, note that disparities for all views are estimated using the single disparity estimator.

AFR: Suppose that the central image I_5 is located on the boundary of LF, *e.g.* the top-left corner in Fig. 2(a). Then, some network branches for adjacent images take zero images as input. The features of those images contain dummy values. For this reason, the conventional algorithms [12], [16] train a different network according to the location of I_5 separately. However, this approach is inefficient in terms of both memory and computations.

To overcome this problem, instead of the separate training, we remix the concatenated feature $\mathbf{S}$ adaptively according to the location of I_5 based on the channel-wise pooling. For efficient remixing, we enforce view-specific constraints, which allow each multi-view feature $\mathbf{S}_i$ to affect $\mathbf{S}_j$ only when I_i and I_j are 8-adjacent. For example, Fig. 3 illustrates the AFR processes for (a) the top-left case and (b) the top case, respectively. In both cases, for instance, $\mathbf{S}_1$ is remixed with the features $\mathbf{S}_2$, $\mathbf{S}_4$, and $\mathbf{S}_5$ of the three adjacent images only.

More specifically, we remix the feature $\mathbf{S}$ to obtain a new feature $\tilde{\mathbf{S}}$. Let $\mathbf{s}$ and $\tilde{\mathbf{s}}$ denote the feature vectors, taken out from $\mathbf{S}$ and $\tilde{\mathbf{S}}$ at a spatial position (x, y). Both $\mathbf{s}$ and $\tilde{\mathbf{s}}$ are column vectors in $\mathbb{R}^{9C}$. Then, the feature remixing can be expressed as a matrix multiplication,

$$\tilde{\mathbf{s}} = (\mathbf{W} \otimes \mathbf{C})\mathbf{s} \tag{5}$$

where $\otimes$ denotes the element-wise multiplication, and $\mathbf{W}$ is a trainable matrix of size $9C \times 9C$. Also, $\mathbf{C}$ is a binary matrix of size $9C \times 9C$, which enforces the aforementioned view-specific constraints. Let us define an indexing function

$$\eta(i) = \lceil i/C \rceil \tag{6}$$

where $\lceil \cdot \rceil$ is the ceiling function. Then, the ith element in $\mathbf{s}$ is a feature extracted from $I_{\eta(i)}$. Let c_{ij} be the (i, j)th element in matrix $\mathbf{C}$. The view-specific constraints are enforced by setting c_{ij} to 1 when $I_{\eta(i)}$ and $I_{\eta(j)}$ are adjacent, and 0 otherwise. The feature remixing in (5) is performed for all spatial positions (x, y) in $\mathbb{N}_W \times \mathbb{N}_H$. Consequently, we obtain the remixed feature $\tilde{\mathbf{S}}$.

The remixing matrix $\mathbf{M} \triangleq \mathbf{W} \otimes \mathbf{C}$ in (5) is trained separately for the nine cases in Fig. 2(a). For example, as in Fig. 3, different matrices $\mathbf{W}^{\mathrm{TL}}$ and $\mathbf{W}^{\mathrm{T}}$ are trained for the top-left case and for the top case, respectively. But, the proposed algorithm needs to train only one spatial SR generator in Fig. 2(c), regardless of the location of the central image I_5. In the test phase, we simply change the remixing matrix $\mathbf{M}$, instead of the entire SR network, according to the location of I_5. In this way, the proposed algorithm can save the memory for network parameters and reduce the training time.

Upsampling: The spatial SR generator in Fig. 2(c) processes the remixed feature $\tilde{\mathbf{S}}$ to produce a high-resolution version $\tilde{I}_5^{\mathrm{HR}}$ of the central image I_5. The generator consists of two convolution layers, two dense blocks [41], three convolution layers, one pixel-shuffle layer [30], and two convolution layers. The first two convolution layers reduce the channel dimension of $\tilde{\mathbf{S}}$ from $9C$ to C. The pixel-shuffle layer increases the spatial resolution from $W \times H$ to $r_s W \times r_s H$.

Learning: We train the spatial SR network by minimizing a loss function

$$\mathcal{L} = \mathcal{L}_{\mathcal{S}} + 0.01\mathcal{L}_{\mathcal{W}} + 0.01\mathcal{L}_{D} \tag{7}$$

where $\mathcal{L}_{\mathcal{S}}$ is the mean squared error between a spatial SR result $\hat{I}_{\mathbf{u}}^{\mathrm{HR}}$ and its ground-truth $I_{\mathbf{u}}^{\mathrm{HR}}$. The warping loss $\mathcal{L}_{\mathcal{W}}$ improves

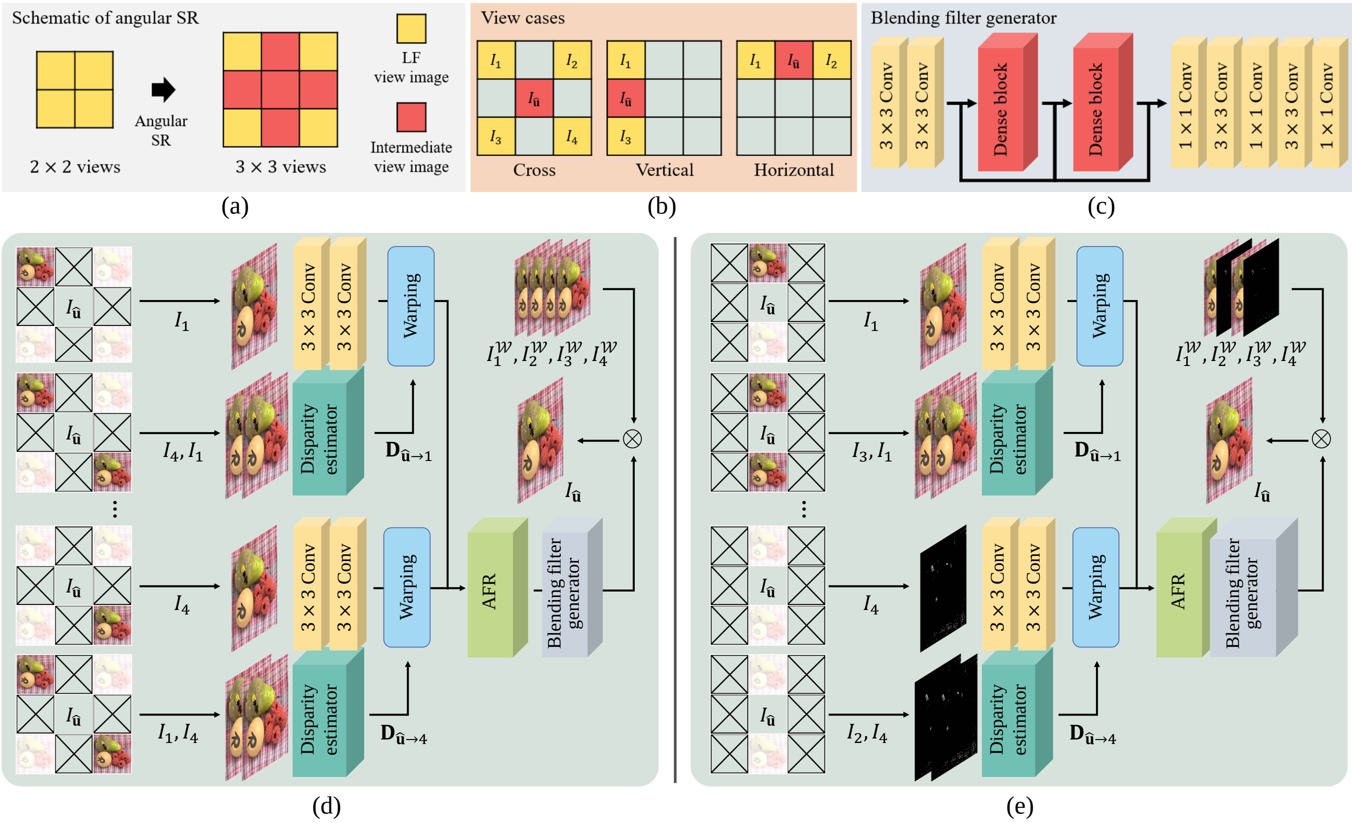


Fig. 4: (a) Generating five intermediate images between four LF images, (b) three cases for the angular SR, and (c) the blending filter generator, (d) the angular SR in the cross case, and (e) the angular SR in the vertical case.

the accuracy of the disparity estimation, by penalizing the error between the central image with warped adjacent images;

$$\mathcal{L}_{\mathcal{W}} = \frac{1}{8}\sum_{i=1, i\neq 5}^{9} \left\| I_5 - \mathcal{W}(I_i, \mathbf{D}_{5\to i}) \right\|_1 . \tag{8}$$

The disparity smoothness loss $\mathcal{L}_D$ constrains neighboring pixels to have similar disparities;

$$\mathcal{L}_D = \frac{1}{8}\sum_{i=1, i\neq 5}^{9} \left\| \nabla \mathbf{D}_{5\to i} \right\|_1 . \tag{9}$$

Every component of the proposed network is differentiable. Therefore, we perform the end-to-end training.

B. Angular SR Network

Fig. 4(a) illustrates how to generate five intermediate images between four LF images to increase the angular resolution. Let I_1, I_2, I_3, and I_4 denote those four images, as shown in Fig. 4(b). Also, let $I_{\hat{\mathbf{u}}}$ denote one of the intermediate images. There are three cases: ‘cross,’ ‘vertical,’ and ‘horizontal.’ In the cross case, all reference images, $\{I_i\}_{i=1}^4$, are fed into the angular SR network, as shown in Fig. 4(d). In the vertical case, two reference images (I_1 and I_3) and two zero images are fed into the network, as in Fig. 4(e). The horizontal case is similar to the vertical one. The angular SR network performs multi-view feature extraction and AFR similarly to the spatial SR network. However, it uses the blending filter generator, instead of the spatial SR generator.

Multi-view feature extraction: We extract multi-view features by feeding four reference images to different network branches. Then, we warp the extracted feature of each reference image I_i to the intermediate image $I_{\hat{\mathbf{u}}}$ using a disparity map $\mathbf{D}_{\hat{\mathbf{u}}\to i}$. Since there is no image information about $I_{\hat{\mathbf{u}}}$, we approximately estimate the disparity $\mathbf{D}_{\hat{\mathbf{u}}\to i}$ using the disparity between the two reference images I_i and I_j, which are symmetrically located with respect to $I_{\hat{\mathbf{u}}}$. For example, in the cross case, two pairs (I_1, I_4) and (I_2, I_3) are used to approximate the disparity information. Specifically, $\mathbf{D}_{\hat{\mathbf{u}}\to i}$ is approximated by

$$\mathbf{D}_{\hat{\mathbf{u}}\to i} = 0.5\mathbf{D}_{j\to i}. \tag{10}$$

Then, we obtain spatially aligned features by warping the extracted features with the approximated disparities and then concatenate the aligned features. Note that the angular SR network uses the same disparity estimator (with the same parameters) as the spatial SR network does.

AFR: To handle all three cases in Fig. 4(b) using the single angular SR network, we adopt AFR with the remixing matrices of size $4C \times 4C$. By varying the remixing matrix according to an angular position, we obtain the remixed feature for the corresponding intermediate image.

TABLE I: Comparison of the proposed algorithm with the conventional algorithms in terms of PSNR/SSIM scores for scale factor $\times 2$ and for all view images. The best results are boldfaced, and the second best ones are underlined.

	Datasets					
	HCI [18]	HCI2 [20]	EPFL [19]	Bikes [21]	Occlusions [21]	Reflective [21]
Bicubic	35.23/0.930	31.67/0.882	31.23/0.886	29.76/0.901	33.60/0.927	36.94/0.950
LFNet [15]	36.46/0.964	33.63/0.932	32.70/0.935	31.92/0.950	35.92/0.963	38.80/0.971
EDSR [31]	39.24/0.966	35.07/0.949	33.94/0.947	33.86/0.964	37.61/0.969	40.64/0.976
SOF-VSR [42]	39.12/0.959	34.75/0.932	34.61/0.934	33.52/0.951	37.64/0.962	40.49/0.969
ResLF [16]	41.09/0.988	36.45/0.979	35.48/0.973	35.21/0.981	39.71/0.988	42.32/0.990
Proposed	**42.06/0.989**	**37.27/0.980**	**37.21/0.977**	**36.00/0.982**	**40.24/0.988**	**42.77/0.991**

TABLE II: Comparison of the proposed algorithm with the conventional algorithms in terms of PSNR/SSIM scores for scale factor $\times 4$ and for central view images.

	Datasets					
	HCI [18]	HCI2 [20]	EPFL [19]	Bikes [21]	Occlusions [21]	Reflective [21]
RCAN [43]	34.10/0.883	30.29/0.810	28.45/0.799	27.65/0.833	31.41/0.868	35.36/0.916
SRFBN [35]	34.09/0.883	29.92/0.809	28.71/0.800	27.64/0.832	31.42/0.868	35.38/0.917
SOF-VSR [42]	32.78/0.863	28.86/0.782	29.32/0.794	26.66/0.801	30.45/0.849	34.09/0.905
ResLF [16]	34.40/0.951	30.25/0.913	27.89/0.895	27.61/0.906	32.00/0.943	35.41/0.963
Proposed	**34.98/0.956**	**31.45/0.926**	**31.48/0.916**	**29.36/0.919**	**33.33/0.948**	**36.69/0.965**

Blending: We reconstruct each intermediate image $I_{\hat{\mathbf{u}}}$, by superposing warped reference images $\{I_i^{\mathcal{W}}\}_{i=1}^4$ with blending filters, where

$$I_i^{\mathcal{W}} = \mathcal{W}(I_i, \mathbf{D}_{\hat{\mathbf{u}} \to i}). \tag{11}$$

Using the remixed feature, the blending filter generator in Fig. 4(c) yields a feature of size $H \times W \times 36$. Then, the feature is split into $3 \times 3 \times 4$ data for each pixel position (x, y), denoted by $\mathbf{B}_{x,y} \in \mathbb{R}^{3 \times 3 \times 4}$. We use $\mathbf{B}_{x,y}$ as the dynamic blending filter [44]. More specifically, we reconstruct the intermediate image by

$$\tilde{I}_{\hat{\mathbf{u}}}(x, y) = \sum_{i=1}^{4} \sum_{m=-1}^{1} \sum_{n=-1}^{1} \mathbf{B}_{x,y}(m, n, i) I_i^{\mathcal{W}}(x+m, y+n). \tag{12}$$

Thus, by generating the filter coefficients dynamically, we blend local information in the four warped images effectively and yield a faithfully reconstructed image.

Learning: We use the same disparity estimator, trained for the spatial SR network. We train the other parts of the angular SR network, by minimizing the mean square error between an estimated result $\tilde{I}_{\hat{\mathbf{u}}}$ and the ground-truth $I_{\hat{\mathbf{u}}}$.

C. Implementation Details

In each convolution layer, we perform zero padding, and use the leaky rectified linear unit [45] with the slope of 0.2 for negative input as the activation function. We use the Adam optimizer [46] with a learning rate of 10^{-4}. The training is iterated for 1,200,000 batches, each of which includes two sets. For spatial SR, a set consists of 3×3 view images. For angular SR, it consists of 2×2 images. We describe the network architecture in detail in the Appendix A.

IV. Experimental Results

We first compare the proposed spatial SR network with conventional algorithms, including the state-of-the-arts [16], [37]. Second, we assess the proposed angular SR network. Third, we evaluate the performance of joint spatial and angular SR. Fourth, we conduct ablation studies to analyze the proposed networks. Finally, we test the proposed spatial SR network in real applications. For quantitative assessment, we employ the PSNR and SSIM metrics.

A. Assessment for Spatial SR

We train the proposed network in two different settings for fair comparisons with Zhang *et al*. [16] and Yeung *et al*. [37], which use different datasets to train their networks.

Comparison with ResLF [16]: For this comparison, we adopt the same training and test sets as [16], which were collected from the synthetic datasets [18], [20] and the real-world datasets [19], [21]. The training and test sets contain 246 and 46 LF images, respectively. All LF images are cropped to the center 9×9 view images. We generate low-resolution LF images using the bicubic interpolation as specified in [16]. The low-resolution images are super-resolved, and the PSNR/SSIM scores of the super-resolved images are computed against the original (or ground-truth) high-resolution images.

Tables I and II compare the proposed spatial SR network with the conventional LFSR algorithms (LFNet [15] and ResLF [16]), the video SR algorithm (SOF-VSR [42]), and the single-image SR algorithms (EDSR [31], RCAN [43], and SRFBN [35]). The scores are the mean PSNR/SSIM. In Table I, the scores of the conventional algorithms, excluding SOF-VSR, are from [16]. They are the results for all 9×9 view images. In Table II, the scores are obtained from central

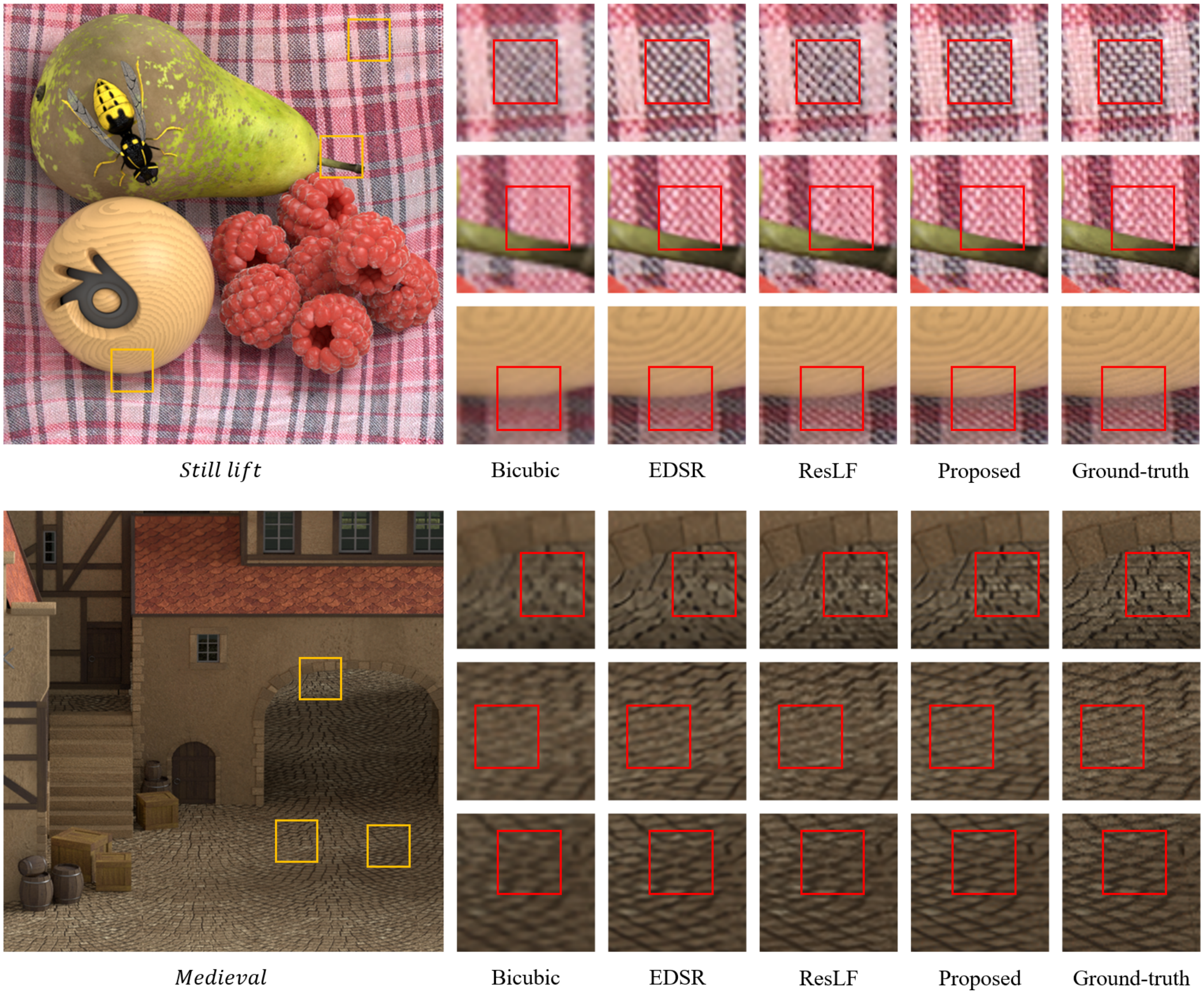


Fig. 5: Qualitative comparison of the proposed spatial SR network with the conventional EDSR [31] and ResLF [16] for scale factor $\times 2$.

TABLE III: Comparison of the proposed algorithm with the conventional LFSR algorithms in terms of PSNR/SSIM scores for scale factors $\times 2$ and $\times 4$.

Scale	Bicubic	LFCNN [12]	RR [36]	GB [11]	LFSSR-SAS [37]	LFSSR-4D [37]	Proposed
$r_s = 2$	34.63/0.935	35.51/0.945	34.18/0.927	35.32/0.944	40.37/0.977	<u>40.67</u>/<u>0.978</u>	**41.26/0.988**
$r_s = 4$	29.41/0.813	30.06/0.828	29.73/0.823	29.99/0.832	33.59/0.910	<u>34.27</u>/<u>0.920</u>	**34.57/0.953**

view images using the source code provided by the authors of each algorithm, because ResLF [16] only provides the model trained for central view images and scale factor $\times 4$.

We see that the proposed algorithm outperforms the state-of-the-art ResLF significantly on all datasets. Fig. 5 and Fig. 6 compare qualitative spatial SR results for scale factor $\times 2$ and $\times 4$, respectively. The proposed algorithm generates less artifacts and provides more faithful images than EDSR, SRFBN, and ResLF do. Note that the proposed algorithm yields high-quality SR results even for the complicated patterns within the red squares.

Comparison with LFSSR [37]: In this test, we use the same dataset and training strategy as in [37]. More specifically, the training and test sets contain 130 and 57 LF images from the Stanford data [21], respectively. The dimension of these LF images is $541 \times 376 \times 8 \times 8$. Only Y color components are used. For generating low-resolution LF images, the images are spatially blurred and then decimated by scale factor r_s.

We compare the proposed algorithm with the conventional algorithms [11], [12], [36], [37] at two scale factors ($\times 2$ and $\times 4$). Table III reports the average PSNR and SSIM scores over all 8×8 view images for all scenes. The scores of

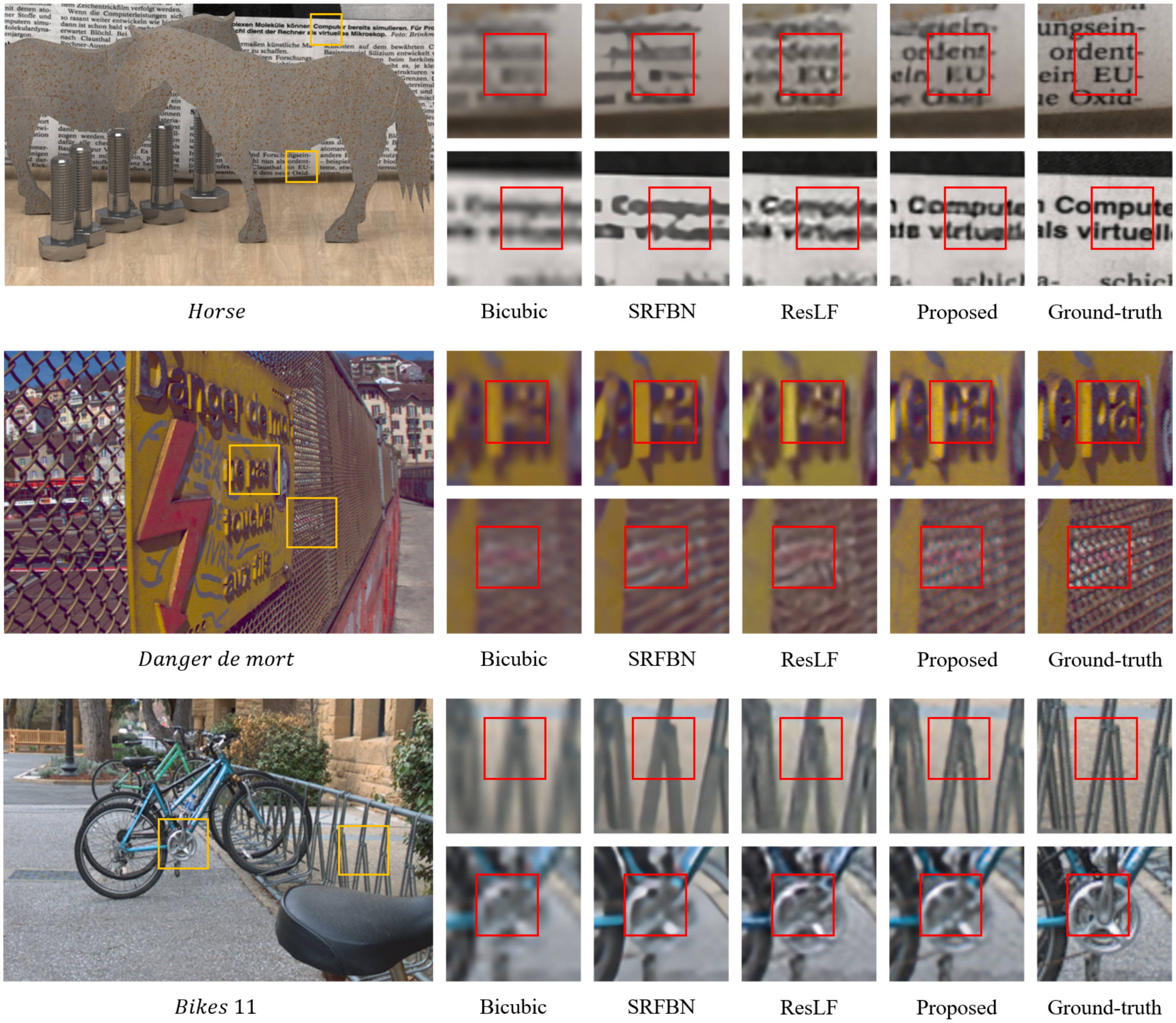


Fig. 6: Qualitative comparison of the proposed spatial SR network with the conventional SRFBN [35] and ResLF [16] for scale factor $\times 4$.

the conventional algorithms are from [37]. LFSSR has two versions: 1) 4D and 2) SAS. Specifically, LFSSR proposes the 4D LFSR network (4D) based on 4D convolution. Then, it reduces the running time by approximating the 4D convolution with 2D convolution, which is called SAS. Notice that the proposed algorithm outperforms both versions, as well as the other conventional algorithms.

Parameters and runtimes: Table IV lists the numbers of parameters and the runtimes of the proposed algorithm and the state-of-the-arts [16], [37]. ResLF [16] uses a large number of parameters, since it includes several networks trained separately according to the angular coordinates. The proposed algorithm achieves the second best performances in terms of both memory and computational complexities. From Tables III and IV, we observe that the proposed algorithm is slower than SAS but outperforms it with meaningful margins bigger than 1dB. Also, note that the proposed algorithm outperforms 4D in terms of both speed and performance.

TABLE IV: Comparison of parameter numbers and execution times. Here, we super-resolve $188 \times 270 \times 8 \times 8$ LF data to $376 \times 540 \times 8 \times 8$. The execution times are measured with a 1080 Ti GPU.

	ResLF [16]	4D [37]	SAS [37]	Proposed
Parameter/runtime	8.0M/2.71s	3.4M/12.1s	**0.8M/1.45s**	<u>1.6M</u>/<u>2.45s</u>

B. Assessment for Angular SR

We evaluate the proposed angular SR algorithm with [3], [47], [48] in Table V. For this comparison, we reduce the angular resolution from 9×9 to 3×3. To reconstruct 9×9 view images, we employ the network twice. Specifically,

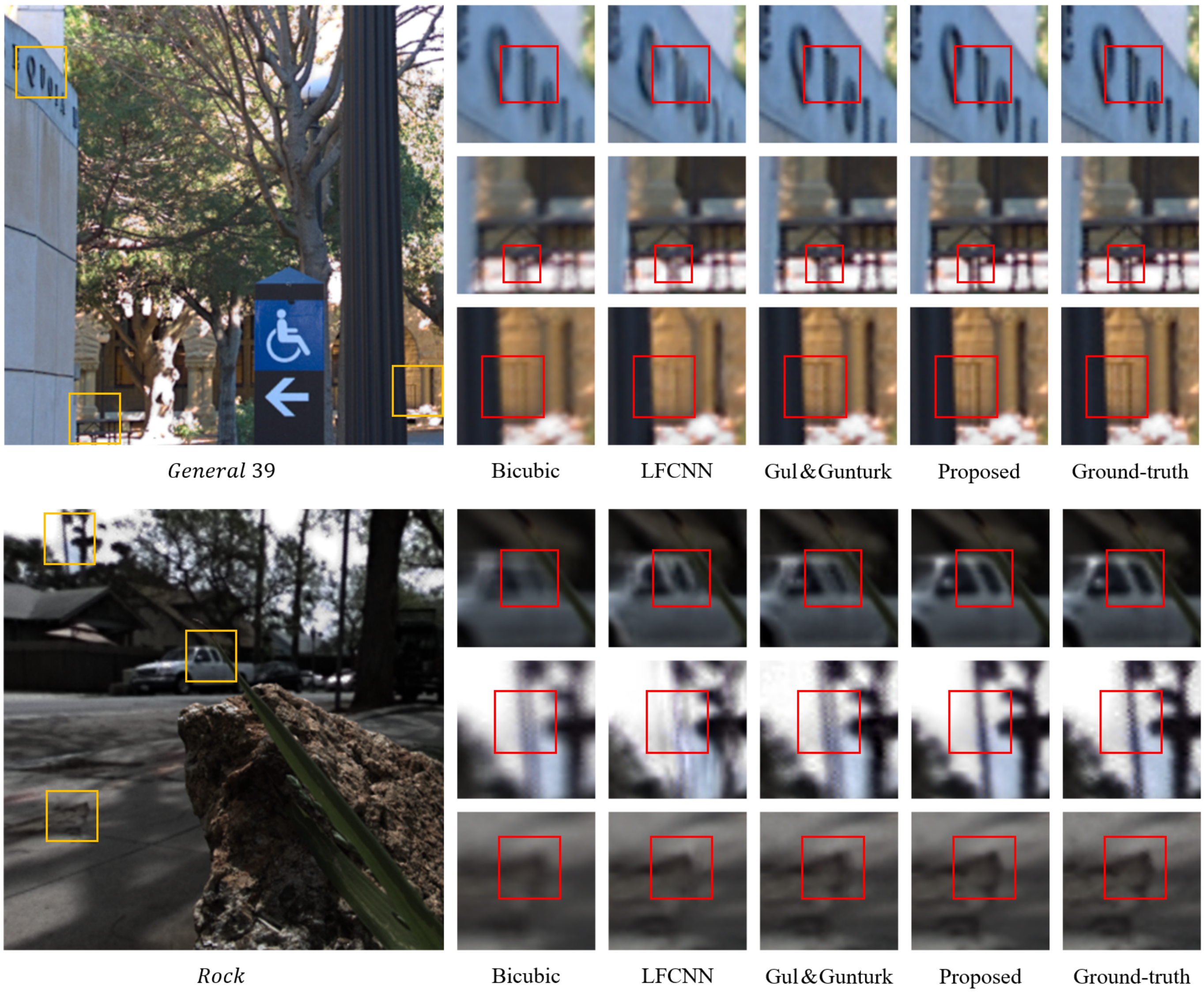


Fig. 7: Qualitative comparison of the proposed angular SR network with the conventional algorithms [12], [14] on the real-world images 'General 39' and 'Rock.'

TABLE V: Comparison of the proposed algorithm with the angular SR algorithms [3], [47], [48] in terms of PSNR scores for the task of $3 \times 3 \rightarrow 9 \times 9$. The best results are boldfaced.

	Buddha	Mona	Average
Kalantari *et al.* [47]	43.20	44.37	43.79
Wu *et al.* [3]	42.73	42.42	42.58
Wing *et al.* [48]	43.77	45.67	44.72
Proposed	**44.38**	**47.74**	**46.06**

we reconstruct the 5×5 view images from 3×3 ones with the proposed algorithm and then, reconstruct the 9×9 view images from the reconstructed 5×5 ones. For training, we use the same training set as [48]. The scores of the conventional algorithm are from [48]. We see that the proposed algorithm outperforms the state-of-the-art [48] significantly on both the 'Buddha' and 'Mona' scenes in the HCI dataset [18].

Table VI compares the proposed angular SR algorithm with the state-of-the-arts algorithms, LFCNN [12] and the Gul and Gunturk's algorithm [14]. The test scenes in [16] are used for this comparison. For the proposed algorithm and LFCNN, we reduce the angular resolution of an original LF image from 9×9 to 5×5 by removing even columns and even rows. Then, we reconstruct the 9×9 views from the reduced LF image. We implemented LFCNN for comparison, since its source codes are unavailable. For a fair comparison, we implemented it to have a similar number of parameters to the proposed algorithm. For training, we use the same training set as [16]. The implementation details for this reproduced LFCNN are available in the Appendix B. The Gul and Gunturk's algorithm [14] is designed to take 7×7 view images to produce 14×14 images. Thus, we reconstruct 14×14 view images using the source code provided by [14], and then crop 9×9 images from the results for the comparison. Notice that [14] cannot provide angular SR results on HCI, HCI2, and EPFL, since it does not support the angular resolutions in these datasets. In Table VI, we see that the proposed algorithm

TABLE VI: Comparison of the proposed algorithm with the reproduced LFCNN and the Gul and Gunturk's algorithm in terms of PSNR/SSIM scores for the angular SR. The best results are boldfaced.

	Datasets					
	HCI [18]	HCI2 [20]	EPFL [19]	Bikes [21]	Occlusions [21]	Reflective [21]
LFCNN [12]	40.32/0.975	35.85/0.938	40.69/0.994	36.96/0.986	38.15/0.983	42.92/0.990
Gul and Gunturk [14]	-	-	-	39.16/0.987	41.15/0.980	43.95/0.981
Proposed	**45.36/0.993**	**39.35/0.980**	**43.40/0.999**	**39.99/0.994**	**42.64/0.993**	**45.59/0.995**

TABLE VII: Quantitative assessment for joint spatial and angular SR. For each test, PSNR/SSIM scores are reported.

Methods	PSNR					
	Buddha			Mona		
	Min	Avg	Max	Min	Avg	Max
Mitra and Veeraraghavan [9]	22.61/0.611	26.76/0.776	32.37/0.913	24.36/0.633	28.11/0.773	34.53/0.956
Wanner and Goldluecke [49]	21.77/0.525	25.50/0.650	33.83/0.911	25.46/0.598	29.62/0.743	36.84/0.944
Bicubic	34.22/0.925	34.63/0.933	35.14/0.947	34.10/0.948	34.20/0.950	34.25/0.951
Spatial (Bicubic)→Angular (LFCNN)	35.68/0.928	35.79/0.929	35.87/0.930	35.80/0.936	35.91/0.936	35.99/0.937
Spatial (Bicubic)→Angular (Proposed)	36.53/0.969	37.19/0.973	37.78/0.976	37.16/0.978	37.49/0.980	37.73/0.981
Angular (LFCNN)→Spatial (LFCNN)	36.54/0.955	36.64/0.956	36.71/0.957	37.10/0.966	37.20/0.966	37.28/0.966
Angular (Proposed)→Spatial (Proposed)	**38.41/0.978**	**40.17/0.986**	**41.23/0.989**	**42.63/0.992**	43.11/0.993	43.50/0.993
Spatial (LFCNN)→Angular (LFCNN)	35.76/0.947	35.87/0.948	35.93/0.948	36.25/0.958	36.33/0.958	36.39/0.958
Spatial (Proposed)→Angular (Proposed)	38.19/0.978	39.69/0.984	40.80/0.989	42.55/0.992	**43.55/0.994**	**44.26/0.994**

TABLE VIII: Expanded test set for ablation studies. The scenes in the boldfaced fonts are newly included, while the others are in the original test set in [16]. The expanded test set contains 76 scenes in total.

Dataset (# scenes)	Scenes				
HCI (9)	*Buddha* ***Medieval***	*Horses* ***Elephant***	*Mona* ***Watch***	*Papillon* ***Still Lift***	***Cone Head***
HCI2 (4)	*Bedroom*	*Bicycle*	*Boxes*	*Sideboard*	
EPFL (13)	*Flowers* *Reeds* ***Magnets 1***	*Friends 5* *University* ***Vespa***	*Fountain Pool* *Paved Road* ***Fountain & Vincent 2***	*ISO Chart 12* ***Color chart 1***	*Palais du Luxembourg* ***Ankylosaurus & Diplodocus 1***
Stanford (50)	*Bikes 1-10*	*Occlusions 1-10*	*Reflective 1-10*	***Buildings 1-10***	***Cars 1-10***

outperforms LFCNN and [14] with large margins.

Fig. 7 compares reconstructed intermediate images of the proposed algorithm with those of the conventional algorithms [12], [14] on the 'General 39' scene in the Stanford dataset [21] and the 'Rock' scene in [47]. Whereas the conventional algorithms [12], [14] produce noticeable artifacts and blurred edges, the proposed algorithm reconstructs sharp and clear edges. Especially, within the red square regions, the conventional algorithms fail to reconstruct object shapes, but the proposed algorithm yields faithful results.

C. Assessment for Joint Spatial and Angular SR

We analyze the performance of joint spatial and angular SR in Table VII. We reduce the angular resolution from 9×9 to 5×5 and down-sample the spatial resolution with a factor of 2 using the bicubic interpolation. To reconstruct those 9×9 view images with the original spatial resolution, we can first perform spatial SR to increase the spatial resolution of the 5×5 view images, and then do angular SR. Alternatively, we can first perform angular SR and then do spatial SR. We use the 'Buddha' and 'Mona' scenes in the HCI dataset [18] as the test set and the remaining ten scenes as the training set, as done in [12]. Table VII shows the minimum, average, and maximum scores of the reconstructed 9×9 images in terms of PSNR and SSIM. In Table VII, the scores of the conventional algorithms are from [12]. We observe that the proposed algorithm outperforms LFCNN [12] significantly in both methods 'Spatial→Angular' and 'Angular→Spatial.'

In the proposed algorithm, the two methods 'Spatial→Angular' and 'Angular→Spatial' yield similar scores to each other on average. In this particular test, 'Angular→Spatial' performs better on 'Buddha,' while 'Spatial→Angular' on 'Mona.' However, 'Spatial→Angular' requires higher computational complexity than 'Angular→Spatial,' because the angular SR network in 'Spatial→Angular' should take super-resolved images as input. Thus, considering their similar performances, 'Angular→Spatial' is a computationally more efficient choice between the two methods.

TABLE IX: Ablation studies on the proposed spatial SR network: 'w/o warping' and 'w/o AFR' mean that the disparity-based feature warping and the proposed AFR are not used, respectively.

Settings	Datasets				
	HCI [18]	HCI2 [20]	EPFL [19]	Stanford [21]	Average
w/o warping	41.14/0.978	36.58/0.976	36.34/0.980	39.91/0.974	39.27/0.976
w/o AFR	40.38/0.975	35.80/0.968	35.12/0.970	38.74/0.975	38.16/0.974
Proposed	42.17/0.987	37.27/0.980	38.78/0.982	40.20/0.987	40.04/0.986

TABLE X: Ablation studies on the angular SR network.

Settings	I	II	III	IV	V	VI	VII
Output of blending filter generator	Image	Filter	Filter	Filter	Filter	Filter	Filter
Filter size	-	1×1	3×3	5×5	3×3	3×3	3×3
The number of input images	4	4	4	4	2	4	4
AFR	✓	✓	✓	✓	✓		✓
Disparity-based feature warping	✓	✓	✓	✓	✓	✓	
PSNR	42.06	43.41	**43.50**	43.20	42.22	42.02	42.49

D. Ablation Studies

For more reliable ablation studies, we expand the test set in [16]. Whereas [16] uses 46 scenes from HCI [18], [20], EPFL [19], and Stanford [21], we select 30 more scenes from those datasets. Thus, the expanded test set includes 76 scenes in total. Table VIII lists these scenes.

Spatial SR: We analyze the efficacy of each component of the proposed spatial SR network through two ablation studies. First, we measure the performance of the spatial SR network without the disparity-based feature warping. Second, we do not perform AFR. Let us refer to these settings as 'w/o warping,' and 'w/o AFR.' Table IX shows the average PSNR and SSIM scores. Without the feature warping or remixing, the SR performance is degraded severely. This indicates that both the feature warping and the feature remixing are essential components of the proposed algorithm.

Angular SR: We conduct ablation studies for the proposed angular SR network. We test various settings. First, we make the blending filter generator to output an intermediate image directly, instead of the filter $\mathbf{B}_{x,y}$ for each pixel position (x, y). Second, we vary the size of the blending filter from 1×1 to 5×5. Third, we reduce the number of input view images from 4 to 2. When this number is 2 in the cross case, only one pair of two symmetric view images, (I_1, I_4) or (I_2, I_3), is fed into the network. Finally, we do not perform AFR.

Note that the proposed blending filter yields better performance than the direct image generation and achieves the best performance with the kernel size of 3×3. Also, the usage of 4 input images provides better performance than that of 2 images. This indicates that a more faithful intermediate image is reconstructed using 4 input images in the cross case. Finally, we can see that AFR significantly improves the angular SR performance.

Disparity estimator: We analyze the effectiveness of the proposed disparity estimator. To this end, we train the spatial SR network in an end-to-end manner, after replacing the proposed disparity estimator with a more sophisticated optical flow estimator, FlowNet-S in [39]. Table XI compares the results. We see that the proposed disparity estimator requires much fewer parameters than FlowNet-S does, at the cost of slightly lower PSNR scores.

TABLE XI: Comparison of the proposed disparity estimator with FlowNet-S [39] in terms of the number of parameters and PSNR scores.

	Parameters	HCI [18]	HCI2 [20]	EPFL [19]
FlowNet-S	38,662,992	42.23	37.31	38.79
Proposed	119,266	42.17	37.27	38.78

Efficacy of AFR: We investigate the impacts of the proposed AFR in detail. For comparison, without using AFR, we train 12 networks separately: 9 for the nine cases in spatial SR in Fig. 2(a) and 3 for the three cases in Fig. 4(b) in angular SR. These multiple networks are compared with the proposed spatial and angular SR networks. For a fair comparison, we train all networks for the same number of epochs. Table XII compares the mean PSNR/SSIM scores. While the proposed networks yield slightly lower scores than the multiple networks in many cases, the proposed networks are also slightly better in some cases; both approaches are comparable in terms of SR performance. However, the proposed AFR reduces training time and saves the memory for network parameters significantly.

E. Real Applications

We test the proposed algorithm on actual images captured by a multi-view camera in Fig. 8. In this test, we obtain 25 view images of size 640×480 using a 5×5 view camera, and then super-resolve the central view image with scale factor $\times 3$. It is observed that the proposed spatial SR network provides

TABLE XII: Unlike the proposed networks using AFR, multiple networks are trained separately according to the angular coordinates of view images. Their SR results are compared with those of the proposed networks. In each case, the higher score is underlined.

Cases	Network type	Datasets				
		HCI [18]	HCI2 [20]	EPFL [19]	Stanford [21]	Average
Spatial SR network						
Top-left	Multiple networks	41.17/0.984	36.73/0.977	38.28/0.979	39.35/0.985	39.27/0.983
	Proposed network	40.98/0.984	36.27/0.976	38.36/0.979	38.62/0.983	38.73/0.982
Top	Multiple networks	41.96/0.987	37.33/0.979	38.89/0.981	40.14/0.988	39.99/0.986
	Proposed network	41.68/0.986	37.06/0.979	38.64/0.981	40.28/0.988	39.99/0.986
Top-right	Multiple networks	41.20/0.984	36.77/0.977	38.21/0.980	39.97/0.987	39.65/0.985
	Proposed network	41.00/0.984	36.38/0.976	38.43/0.980	39.98/0.987	39.65/0.985
Left	Multiple networks	42.02/0.986	37.25/0.980	38.83/0.982	40.15/0.988	39.99/0.986
	Proposed network	41.92/0.986	36.92/0.979	38.52/0.981	39.99/0.988	39.80/0.986
Middle	Multiple networks	42.51/0.988	37.58/0.982	39.20/0.983	40.50/0.987	40.36/0.986
	Proposed network	42.45/0.988	37.52/0.981	38.88/0.982	40.24/0.987	40.13/0.986
Right	Multiple networks	42.00/0.986	37.22/0.980	38.80/0.981	40.22/0.988	40.03/0.986
	Proposed network	41.90/0.986	36.95/0.979	38.65/0.981	40.35/0.988	40.06/0.986
Bottom-left	Multiple networks	41.22/0.985	36.71/0.977	38.24/0.979	39.92/0.987	39.62/0.985
	Proposed network	41.08/0.984	36.30/0.976	38.47/0.980	39.81/0.987	39.55/0.985
Bottom	Multiple networks	42.08/0.987	37.28/0.980	38.96/0.981	40.19/0.987	40.05/0.986
	Proposed network	41.78/0.986	37.02/0.979	38.74/0.981	40.20/0.987	39.97/0.986
Bottom-right	Multiple networks	41.24/0.984	36.75/0.977	38.37/0.979	40.15/0.988	39.80/0.985
	Proposed network	41.16/0.984	36.30/0.976	38.50/0.980	40.18/0.988	39.81/0.985
Angular SR network						
Cross	Multiple networks	44.61/0.994	41.34/0.982	41.20/0.996	43.09/0.995	42.86/0.994
	Proposed network	44.21/0.993	41.33/0.992	41.39/0.996	43.01/0.994	42.79/0.994
Vertical	Multiple networks	45.63/0.994	41.68/0.992	44.22/0.997	44.25/0.995	44.27/0.995
	Proposed network	44.89/0.991	41.55/0.992	43.85/0.997	44.07/0.995	44.00/0.995
Horizontal	Multiple networks	44.02/0.990	42.31/0.993	43.18/0.997	43.77/0.995	43.62/0.995
	Proposed network	44.34/0.991	42.41/0.993	42.93/0.997	43.69/0.995	43.57/0.994

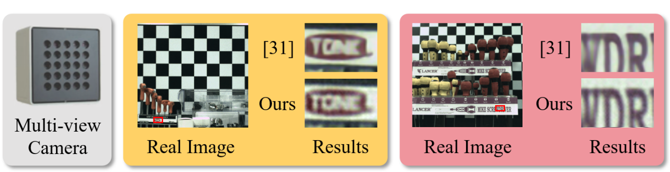


Fig. 8: Comparison of super-resolved images in real applications.

more faithful results especially on the detailed patterns than EDSR [31] does.

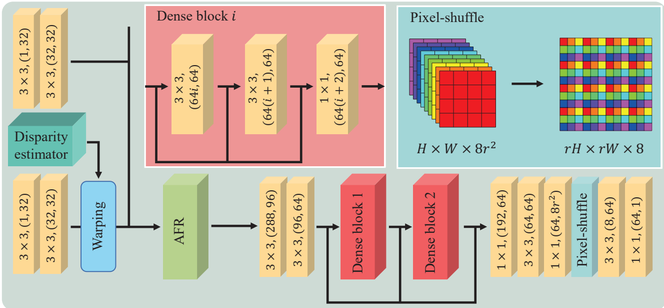
V. Conclusions

In this paper, we developed the LFSR algorithm based on AFR, which yields high quality SR results regardless of the angular coordinates of input views. The proposed spatial and angular SR networks extract multi-view features using the trainable disparity estimator. It then performs the feature remixing according to the angular coordinates and reconstructs images from the remixed features. Experimental results demonstrated that the proposed algorithm outperforms the state-of-the-art algorithms on various datasets.

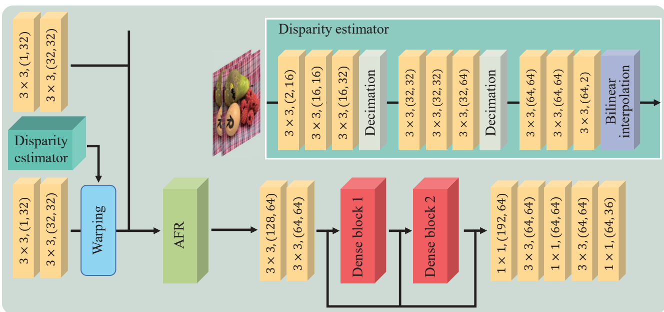
Appendix A
Detailed Network Architecture

Fig. 9 shows the detailed structures of the proposed spatial and angular SR networks. Each convolution layer is labeled as '$k\times k, (c_1, c_2)$,' where k is the kernel size, and c_1 and c_2 are the number of input and output channels, respectively. We perform zero padding and adopt the leaky rectified linear unit [45] with the slope of 0.2 for negative input as the activation function in all convolution layers.

In the spatial SR network, to extract multi-view features, we implement each branch using two convolution layers both with 32 filters. The disparity estimator consists of three convolution blocks. Each convolution block contains three sequential convolution layers. In the first two blocks, the last convolution layers halve the spatial resolutions both horizontally and vertically with stride 2. In the up-sampling step, there are 7 convolution layers, 2 dense blocks [41], and 1 pixel-shuffle layer [30]. Each dense block is composed of three convolution layers. All convolution layers in the dense blocks have 64 filters. The pixel-shuffle layer is a periodic shuffling operator that rearranges an $H\times W\times r_s^2 C$ tensor into an $r_s H\times r_s W\times C$



(a) Spatial SR network



(b) Angular SR network

Fig. 9: Detailed structures of the proposed spatial and angular SR networks.

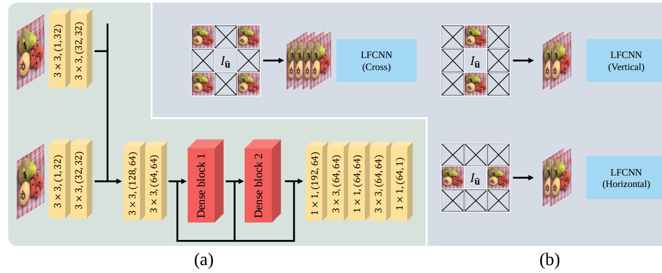


Fig. 10: (a) Reproduced LFCNN. (b) Three LFCNNs for cross, vertical, and horizontal cases.

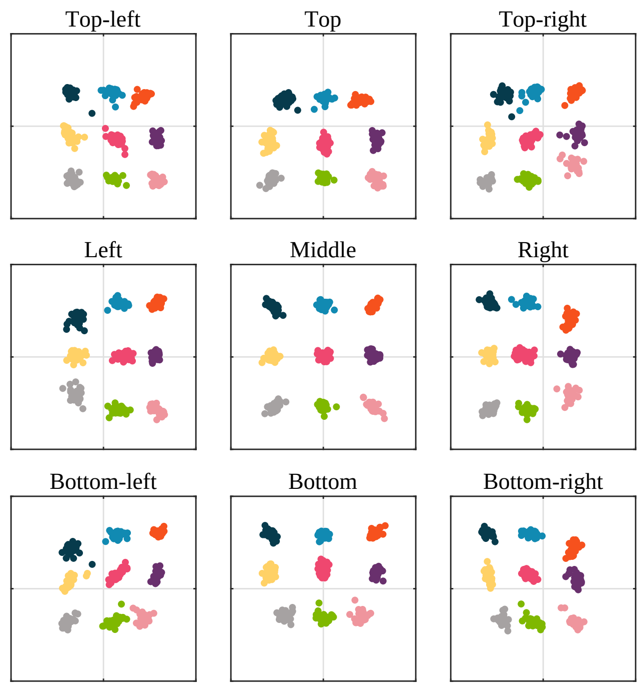


Fig. 11: Visualization of AFR.

tensor, where r_s is the scale factor.

In the angular SR network, to extract multi-view features, we implement the four branches, each of which has two convolution layers both with 32 filters. We use the same disparity estimator trained for the spatial SR network. For blending, we use 7 convolution layers and 2 dense blocks [41]. The structure of these dense blocks is identical with that for the spatial SR network.

Appendix B
Architecture of Reproduced LFCNN

For comparison, we reproduce the LFCNN algorithm based on the proposed angular SR network. By comparing Fig. 10(a) to Fig. 9(b), we see that the feature warping and AFR are removed in the reproduced LFCNN, and the number of filters in the last convolution layer is modified to generate intermediate view images directly. Thus, it has a similar number of parameters to the proposed angular SR network. Also, as in [12], we train three LFCNN networks separately for cross, vertical, and horizontal cases in Fig. 10(b). Table VI confirms that the proposed angular SR network based on feature warping, AFR, and blending is superior to the LFCNN algorithm based on image stack and generation.

Appendix C
Visualization of AFR.

Let us visualize the remixing matrices $\mathbf{M}$ for the nine cases in Fig. 2(a). In (5), the ith element $\tilde{s}_i$ in $\tilde{\mathbf{s}}$ is given by

$$\tilde{s}_i = \sum_{j=1}^{9C} m_{ij} s_j. \tag{13}$$

Note that s_j is a feature from $I_{\eta(j)}$. Let $\mathbf{p}_{\eta(j)}$ be the angular position of $I_{\eta(j)}$. Then, by using $|m_{ij}|$ as a weight, we can compute the centroid

$$\tilde{\mathbf{p}}_i = \frac{1}{\sum_{j=1}^{9C} |m_{ij}|} \sum_{j=1}^{9C} |m_{ij}| \times \mathbf{p}_{\eta(j)}. \tag{14}$$

Fig. 11 plots these centroids $\tilde{\mathbf{p}}_i$ for $i \in \mathbb{N}_{9C}$. It shows how the centroids are shifted for each of the nine cases in Fig. 2(a). For example, in the top-left case, I_1, I_2, I_3, I_4, I_7 are zero images. Thus, their features are suppressed in the remixing in (13) and the corresponding m_{ij}'s tend to have small magnitudes. Thus, in the computation of the centroids in (14), the angular positions of I_1, I_2, I_3, I_4, I_7 are multiplied by small weights, while those of I_5, I_6, I_8, I_9 by big weights. Therefore, we see that the centroids are shifted to the bottom and to the right. Similarly, in the left case, the centroids are shifted to the right so as not to use the features in I_1, I_4, I_7. In contrast, in the middle case, no such shifts are observed.

References

[1] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field image processing: An overview," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 926–954, 2017.

[2] T. Yang, Y. Zhang, X. Tong, X. Zhang, and R. Yu, "A new hybrid synthetic aperture imaging model for tracking and seeing people through occlusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1461–1475, 2013.

[3] G. Wu, M. Zhao, L. Wang, Q. Dai, T. Chai, and Y. Liu, "Light field reconstruction using deep convolutional network on EPI," in *CVPR*, 2017.

[4] H. Zhu, Q. Zhang, and Q. Wang, "4D light field superpixel and segmentation," in *CVPR*, 2017.

[5] R. Raghavendra, K. B. Raja, and C. Busch, "Presentation attack detection for face recognition using light field camera," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 1060–1075, 2015.

[6] "Lytro," https://www.lytro.com/.

[7] "Raytrix. 3D light field camera technology," https://www.raytrix.de/.

[8] T. E. Bishop and P. Favaro, "The light field camera: Extended depth of field, aliasing, and superresolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 5, pp. 972–986, 2011.

[9] K. Mitra and A. Veeraraghavan, "Light field denoising, light field superresolution and stereo camera based refocussing using a GMM light field patch prior," in *CVPRW*, 2012.

[10] 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, 2013.

[11] M. Rossi and P. Frossard, "Graph-based light field super-resolution," in *MMSP*, 2017.

[12] Y. Yoon, H.-G. Jeon, D. Yoo, J.-Y. Lee, and I. S. Kweon, "Learning a deep convolutional network for light-field image super-resolution," in *ICCVW*, 2015.

[13] H. Fan, D. Liu, Z. Xiong, and F. Wu, "Two-stage convolutional neural network for light field super-resolution," in *ICIP*, 2017.

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

[15] Y. Wang, F. Liu, K. Zhang, G. Hou, Z. Sun, and T. Tan, "LFNet: A novel bidirectional recurrent convolutional neural network for light-field image super-resolution," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4274–4286, 2018.

[16] S. Zhang, Y. Lin, and H. Sheng, "Residual networks for light field image super-resolution," in *CVPR*, 2019.

[17] R. A. Farrugia and C. Guillemot, "Light field super-resolution using a low-rank prior and deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.

[18] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4D light fields." in *VMV*, 2013.

[19] M. Řeřábek and T. Ebrahimi, "New light field image dataset," in *QoMEX*, 2016.

[20] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke, "A dataset and evaluation methodology for depth estimation on 4D light fields," in *ACCV*, 2016.

[21] "The Stanford Lytro Light Field Archive," http://lightfields.stanford.edu/LF2016.html/.

[22] R. G. Keys, "Cubic convolution interpolation for digital image processing," *IEEE T. Acoust. Speech.*, vol. 29, no. 6, pp. 1153–1160, 1981.

[23] C. E. Duchon, "Lanczos filtering in one and two dimensions," *J. Appl. Meteorol.*, vol. 18, no. 8, pp. 1016–1022, 1979.

[24] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *ICCV*, 2009.

[25] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Trans. Graph.*, vol. 30, no. 2, p. 12, 2011.

[26] J. Yang, Z. Wang, Z. Lin, S. Cohen, and T. Huang, "Coupled dictionary training for image super-resolution," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3467–3478, 2012.

[27] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *ACCV*, 2014.

[28] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *ECCV*, 2014.

[29] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *CVPR*, 2016.

[30] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *CVPR*, 2016.

[31] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *CVPRW*, 2017.

[32] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *CVPR*, 2018.

[33] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *CVPRW*, 2018.

[34] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun, "Meta-SR: A magnification-arbitrary network for super-resolution," in *CVPR*, 2019.

[35] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu, "Feedback network for image super-resolution," in *CVPR*, 2019.

[36] R. A. Farrugia, C. Galea, and C. Guillemot, "Super resolution of light field images using linear subspace projection of patch-volumes," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 7, pp. 1058–1071, 2017.

[37] H. W. F. Yeung, J. Hou, X. Chen, J. Chen, Z. Chen, and Y. Y. Chung, "Light field spatial super-resolution using deep efficient spatial-angular separable convolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2319–2330, 2019.

[38] M. Levoy and P. Hanrahan, "Light field rendering," in *ACM SIGGRAPH*, 1996.

[39] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *ICCV*, 2015.

[40] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *CVPR*, 2018.

[41] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, 2017.

[42] L. Wang, Y. Guo, L. Liu, Z. Lin, X. Deng, and W. An, "Deep video super-resolution using HR optical flow estimation," *IEEE Trans. Image Process.*, vol. 29, pp. 4323–4336, 2020.

[43] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ECCV*, 2018.

[44] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *NIPS*, 2016.

[45] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML*, 2013.

[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

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

[48] H. Wing Fung Yeung, J. Hou, J. Chen, Y. Ying Chung, and X. Chen, "Fast light field reconstruction with deep coarse-to-fine modeling of spatial-angular clues," in *ECCV*, 2018.

[49] S. Wanner and B. Goldluecke, "Spatial and angular variational super-resolution of 4d light fields," in *ECCV*, 2012.

Keunsoo Ko received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2017, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include image processing and machine learning.

Yeong Jun Koh received the B.S. degree and the Ph.D. degree in electrical engineering from Korea University, Seoul, South Korea, in 2011 and 2018, respectively. As an assistant professor in Mar. 2019, he joined the Department of Computer Science & Engineering, Chungnam National University. His research interests include computer vision and machine learning, especially in the problems of video object discovery and segmentation.

Soonkeun Chang received his B.S. degree in astronomy and space science from KyungHee University, Korea, in 2000 and M.S. degree in control engineering from Kanazawa University in 2003. He received a Ph.D. degree in control engineering from Tokyo Institute of Technology (TITech) in 2007. Now he works at Samsung Electronics Co., Ltd., Korea. His main research interests include computer vision and image processing.

Chang-Su Kim (S'95-M'01-SM'05) received the Ph.D. degree in electrical engineering from Seoul National University with a Distinguished Dissertation Award in 2000. From 2000 to 2001, he was a Visiting Scholar with the Signal and Image Processing Institute, University of Southern California, Los Angeles. From 2001 to 2003, he coordinated the 3D Data Compression Group in National Research Laboratory for 3D Visual Information Processing in SNU. From 2003 and 2005, he was an Assistant Professor in the Department of Information Engineering, Chinese University of Hong Kong. In Sept. 2005, he joined the School of Electrical Engineering, Korea University, where he is a Professor. His research topics include image processing, computer vision, and machine learning. He has published more than 290 technical papers in international journals and conferences. In 2009, he received the IEEK/IEEE Joint Award for Young IT Engineer of the Year. In 2014, he received the Best Paper Award from Journal of Visual Communication and Image Representation (JVCI). He is a member of the Multimedia Systems & Application Technical Committee (MSATC) of the IEEE Circuits and Systems Society. Also, he is an APSIPA Distinguished Lecturer for term 2017-2018. He served as an Editorial Board Member of JVCI and an Associate Editor of IEEE Transactions on Image Processing. He is a Senior Area Editor of JVCI and an Associate Editor of IEEE Transactions on Multimedia.