Reference-based Video Super-Resolution Using Multi-Camera Video Triplets

Junyong Lee Myeonghee Lee Sunghyun Cho Seungyong Lee
POSTECH
{junyonglee, myeonghee, s.cho, leesy}@postech.ac.kr

Abstract

We propose the first reference-based video super-resolution (RefVSR) approach that utilizes reference videos for high-fidelity results. We focus on RefVSR in a triple-camera setting, where we aim at super-resolving a low-resolution ultra-wide video utilizing wide-angle and telephoto videos. We introduce the first RefVSR network that recurrently aligns and propagates temporal reference features fused with features extracted from low-resolution frames. To facilitate the fusion and propagation of temporal reference features, we propose a propagative temporal fusion module. For learning and evaluation of our network, we present the first RefVSR dataset consisting of triplets of ultra-wide, wide-angle, and telephoto videos concurrently taken from triple cameras of a smartphone. We also propose a two-stage training strategy fully utilizing video triplets in the proposed dataset for real-world 4× video super-resolution. We extensively evaluate our method, and the result shows the state-of-the-art performance in 4× super-resolution.

1. Introduction

Recent mobile devices such as Apple iPhone or Samsung Galaxy series are manufactured with at least two or three asymmetric multi-cameras typically having different but fixed focal lengths. In a triple camera setting, each ultra-wide, wide-angle, and telephoto camera has a different field of view (FoV) and optical zoom factor. One advantage of such configuration is that, compared to an ultra-wide camera, a wide-angle camera captures a subject with more details and higher resolution, and the advantage escalates even further with a telephoto camera. A question naturally follows is why not leverage higher-resolution frames of a camera with a longer focal length to improve the resolution of frames of a camera with a short focal length.

Utilizing a reference (Ref) image to reconstruct a high-resolution (HR) image from a low-resolution (LR) image has been widely studied in previous reference-based image super-resolution (RefSR) approaches [2,23,26,28,29,32,34,35]. However, it has not been explored yet to utilize a Ref video for video super-resolution (VSR). In this paper, we expand the RefSR to the VSR task and introduce reference-based video super-resolution (RefVSR) that can be applied for videos captured in an asymmetric multi-camera setting.

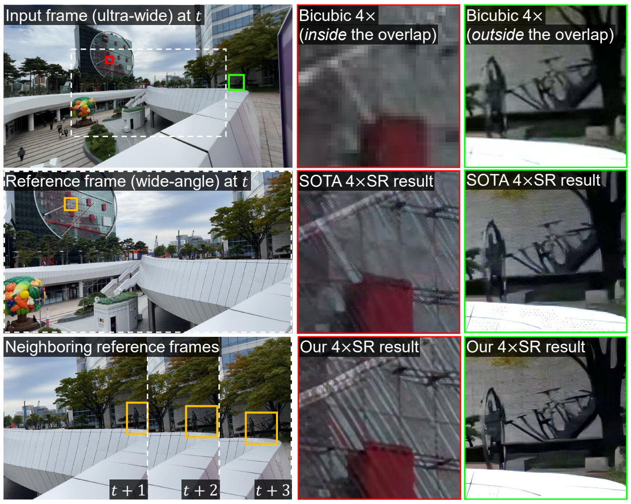

Figure 1. Comparison on 8K 4×SR video results from a real HD video between state-of-the-art (SOTA) RefSR approach [26] and the proposed RefVSR approach. Our method learns to super-resolve an LR video by utilizing relevant high-quality patches of reference frames and robustly recovers sharp textures of both inside and outside the overlapped FoV between the input ultra-wide and reference wide-angle frames (white dashed box).

RefVSR inherits objectives of both RefSR and VSR tasks and utilizes a Ref video for reconstructing an HR video from an LR video. Applying RefVSR for a video captured in an asymmetric multi-camera setting requires consideration of the unique relationship between LR and Ref frames in multi-camera videos. In the setting, a pair of LR and Ref frames at each time step shares almost the same content in their overlapped FoV (top and middle rows of the leftmost column in Fig. 1). Moreover, as a video exhibits a motion, neighboring Ref frames might contain high-quality contents useful for recovering the outside the overlapped FoV (the bottom row of the leftmost column in Fig. 1).

Code and dataset: https://github.com/codeslake/RefVSR

For successful RefVSR in an asymmetric multi-camera setting, we take advantage of temporal Ref frames in reconstructing regions both inside and outside the overlapped FoV. In previous RefSR approaches [26, 28, 29, 32], global matching has been a common choice for establishing non-local correspondence between a pair of LR and Ref images. However, given a pair of LR and Ref video sequences, it is not straightforward to directly apply global matching between an LR frame and *multiple* Ref frames. To utilize as many frames as possible in the global matching for large real-world videos (*e.g.*, HD videos), we need a framework capable of managing Ref frames in a memory-efficient way.

We propose the first end-to-end learning-based RefVSR network that can generally be applied for super-resolving an LR video using a Ref video. Our network adopts a bidirectional recurrent pipeline [4, 7, 8] to recurrently align and propagate Ref features that are fused with the features of LR frames. Our network is efficient in terms of computation and memory consumption because the global matching needed for aligning Ref features is performed only between a pair of LR and corresponding Ref frames at each time step. Still, our network is capable of utilizing temporal Ref frames, as the aligned Ref features are continuously fused and propagated in the pipeline.

As a key component for managing Ref features in the pipeline, we propose a propagative temporal fusion module that fuses and propagates only well-matched Ref features. The module leverages the matching confidence computed during the global matching between LR and Ref features as the guidance to determine well-matched Ref features to be fused and propagated. The module also accumulates the matching confidence throughout the pipeline and uses the accumulated value as the guidance when fusing the propagated temporal Ref features.

To train and validate our model, we present the first RefVSR dataset consisting of 161 video triplets of ultra-wide, wide-angle, and telephoto videos simultaneously captured with triple cameras of a smartphone. Wide-angle and telephoto videos have the same size as ultra-wide videos but their resolutions are 2× and 4× the resolution of ultra-wide videos, respectively. With the RefVSR dataset, we train our network to super-resolve an ultra-wide video 4× to produce an 8K video with the same resolution as a telephoto video. To this end, we propose a two-stage training strategy that fully utilizes video triplets in the proposed dataset. We show that, with our training strategy, our network can successfully learn super-resolution of a real-world HD video and produce a high-fidelity 8K video.

To summarize, our contributions include:

- the first RefVSR framework with the focus on videos recorded in an asymmetric multi-camera setting,
- the propagative temporal fusion module that effectively fuses and propagates temporal Ref features,
- the RealMCVSR dataset, which is the first dataset for the RefVSR task, and
- the two-stage training strategy fully utilizing video triplets for real-world 4×VSR.

2. Related Work

Reference-based Super-Resolution (RefSR) Previous RefSR approaches [2, 23, 26, 28, 29, 32, 34, 35] have focused on establishing non-local correspondence between LR and Ref features. For establishing correspondence, either offset-based matching (optical flow [35] and deformable convolution [23]) or patch-based matching (patch-match [1, 2, 32, 34], learnable patch-match [28, 29], learnable patch-match with affine correction [26]) are employed.

Video Super-Resolution (VSR) Previous VSR methods have focused on how to effectively utilize highly related but unaligned LR frames in a video sequence. With respect to how LR frames in video sequences are handled by a model, previous VSR approaches can be categorized into either sliding window-based [3, 14, 15, 25, 27] or recurrent framework-based [4, 7, 8, 9, 21] approaches. For handling unaligned LR frames, warping using optical flow [3, 4, 21], patch-based correlation [15], and deformable convolution [25, 27] have been employed.

The aforementioned previous studies in RefSR and VSR have developed various components. In this paper, to match and align Ref features to an LR frame, we adopt the learnable patch-match-based reference alignment module [26, 28, 29]. To handle video sequences, we adopt a bidirectional recurrent framework [4, 7, 8]. However, for RefVSR, we modify the components to handle both LR and Ref videos. We also equip our network with the propagative temporal fusion module, designed to effectively and efficiently exploit temporal Ref features in reconstructing HR frames.

Recently, a RefVSR method [33], in which only the first frame of an HR video is used as a reference to super-resolve an LR video downsampled from the HR video, has been concurrently proposed alongside our work. However, to the best of our knowledge, ours is the first RefVSR framework that utilizes multiple frames in a Ref video for super-resolving a real-world LR video.

3. Multi-Camera Video Super-Resolution

3.1. Framework Overview

Fig. 2 shows an overview of the proposed network, which can generally be applied to a RefVSR task for super-resolving an LR video utilizing a Ref video. Our network follows a typical bidirectional propagation scheme [4, 7], consisting of bidirectional recurrent cells F_f and F_b, where the subscripts f and b indicate forward and backward propagation branches, respectively (Fig. 3). Our network is dis-

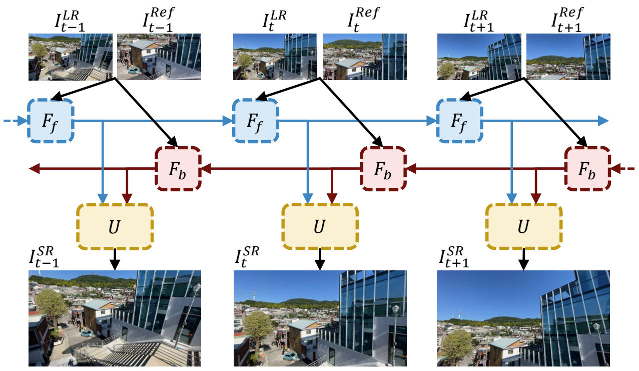

Figure 2. Overview of our RefVSR framework.

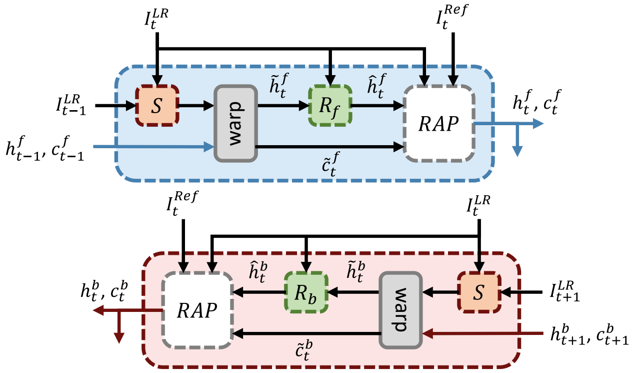

Figure 3. Forward (top) and backward (bottom) recurrent cells.

tinguished from previous ones in additional inputs, intermediate features, and modules to utilize a Ref video sequence.

Specifically, for a time step t, each recurrent cell F_f or F_b takes not only low-resolution LR frames $I_{t\pm1}^{LR}$ at the previous time step and I_t^{LR} at the current time step, but also a Ref frame I_t^{Ref} at the current time step. Each cell is also recurrently fed with aggregated LR and Ref features $h_{t\pm1}^{\{f,b\}}$ and accumulated confidence maps $c_{t\pm1}^{\{f,b\}}$ propagated from the previous time step. Here, the accumulated confidence maps are utilized for fusing well-matched Ref features later in each recurrent cell. Finally, each recurrent cell propagates the resulting features $h_t^{\{f,b\}}$ and the accumulated matching confidences $c_t^{\{f,b\}}$ to the next cell. Formally, we have:

$$\begin{aligned}\{h_t^f, c_t^f\} &= F_f(I_{t-1}^{LR},\, I_t^{LR},\, I_t^{Ref},\, h_{t-1}^f,\, c_{t-1}^f),\\ \{h_t^b, c_t^b\} &= F_b(I_{t+1}^{LR},\, I_t^{LR},\, I_t^{Ref},\, h_{t+1}^b,\, c_{t+1}^b).\end{aligned} \tag{1}$$

For reconstructing an SR result I_t^{SR}, the upsampling module U first takes the intermediate features $h_t^{\{f,b\}}$ and accumulated matching confidences $c_t^{\{f,b\}}$ of both forward and backward branches. Then, the features are aggregated and upsampled with multiple convolution and pixel-shuffle [22] layers to produce I_t^{SR}. Mathematically, we have:

$$I_t^{SR} = U(h_t^f,\, h_t^b,\, c_t^f,\, c_t^b). \tag{2}$$

For the upsampling module U to accurately reconstruct I_t^{SR}, the intermediate features $h_t^{\{f,b\}}$ should contain details integrated from both LR and Ref frames in a video sequence. To this end, each recurrent cell F_f and F_b performs inter-frame alignment between the previous and current LR input frames, then aggregates and propagates the features (Sec. 3.2). To exploit multiple Ref frames, each recurrent cell aligns the current Ref features to the current LR frame and fuses the aligned Ref features to the aggregated features of the previous Ref, LR, and current LR frames using a reference alignment and propagation module (Sec. 3.3). In this way, features of temporally distant LR input and Ref frames can be recurrently integrated and propagated.

3.2. Bidirectional Recurrent Cells

In each recurrent cell F_f and F_b (Fig. 3), we first use a flow estimation network S [20] to estimate the optical flow between the LR frame I_t^{LR} at the current time step and $I_{t\pm1}^{LR}$ at the previous time step to align propagated features $h_{t\pm1}^{\{f,b\}}$ to I_t^{LR}. Then, using a residual block R, we aggregate an LR frame I_t^{LR} into the aligned features to obtain *temporally aggregated features* $\widehat{h}_t^{\{f,b\}}$. Specifically, we have:

$$\begin{aligned} w_t^{\{f,b\}} &= S(I_t^{LR},\, I_{t\pm1}^{LR}),\\ \widetilde{h}_t^{\{f,b\}} &= \mathrm{warp}(h_{t\pm1}^{\{f,b\}},\, w_t^{\{f,b\}}),\\ \widehat{h}_t^{\{f,b\}} &= R_{\{f,b\}}(I_t^{LR},\, \widetilde{h}_t^{\{f,b\}}),\end{aligned} \tag{3}$$

where $\mathrm{warp}(,)$ denotes warping operation, and $w_t^{\{f,b\}}$ is the optical flow estimated by the flow estimation network S. Note that the temporally aggregated features $\widehat{h}_t^{\{f,b\}}$ contains details aggregated from multiple LR features, as well as temporal Ref features propagated from neighboring cells.

Now we propose the reference alignment and propagation module for each cell F_f and F_b to fuse the current Ref frame I_t^{Ref} into temporally aggregated features $\widehat{h}_t^{\{f,b\}}$.

3.3. Reference Alignment and Propagation

Our reference alignment and propagation module (Fig. 4) consists of three sub-modules: cosine similarity, reference alignment, and propagative temporal fusion modules. The cosine similarity module computes a cosine similarity matrix between the Ref frame I_t^{Ref} and target LR frames I_t^{LR} and computes an index map p_t and a confidence map c_t needed for the other two sub-modules. The reference alignment module extracts a feature map from the current Ref frame I_t^{Ref} and warps the feature map to I_t^{LR} using the index map p_t. Then, the propagative temporal fusion module fuses the aligned Ref features with the *temporally aggregated features* $\widehat{h}_t^{\{f,b\}}$. In the following, we describe each module in more detail.

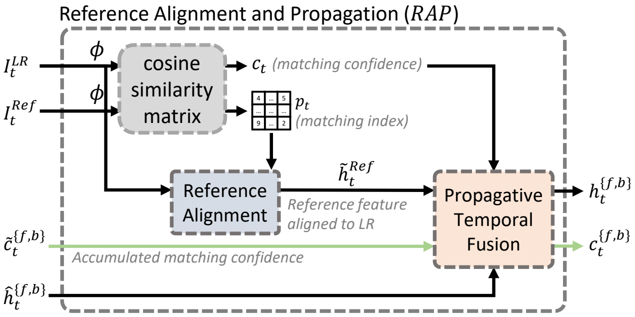

Figure 4. The reference alignment and propagation module.

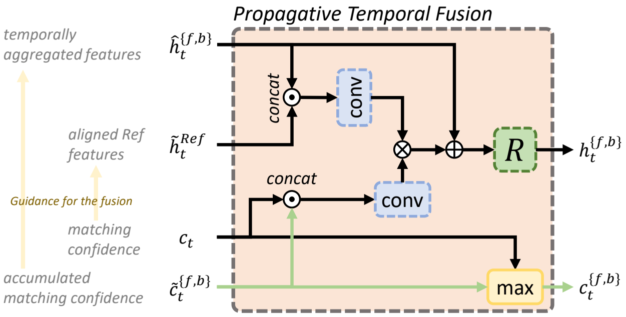

Figure 5. The propagative temporal fusion module.

Cosine Similarity Module To compute an index map p_t and a confidence map c_t, we first embed I_t^{LR} and $I_{t\downarrow}^{Ref}$ into the feature space by a shared encoder ϕ [24], where $\downarrow$ denotes the downsampling operator. Then, we extract 3×3 patches from the LR and Ref feature maps with stride 1 and compute a cosine similarity matrix C between them, such that $C_{i,j}$ is a similarity between the i-th patch of the LR feature map and the j-th patch of the Ref feature map. The matching index map p and confidence map c is then computed as:

$$p_{t,i} = \operatorname*{argmax}_j C_{i,j}, \qquad c_{t,i} = \max_j C_{i,j}, \tag{4}$$

where $p_{t,i}$ is the patch index of Ref features $\phi(I_{t\downarrow}^{Ref})$ that is the most relevant to the i-th patch of LR features $\phi(I_t^{LR})$, and $c_{t,i}$ is their matching confidence, respectively.

Reference Alignment Module We use the reference alignment module proposed in [26] to obtain Ref features aligned to I_t^{LR}, which will be used for the fusion later. The module first takes I_t^{Ref} and extracts Ref features h_t^{Ref}. Then, using the matching index map p_t (Eq. 4), we warp patches of Ref features h_t^{Ref} to coarsely align the features to the current LR frame I_t^{LR} [28,29]. Finally, the module compensates for possible inter-patch misalignment (*e.g.*, scale and rotation) in the coarsely aligned Ref features using the patch-wise affine spatial transformer [5, 10]. We denote the final aligned Ref features as $\widetilde{h}_t^{Ref}$.

Propagative Temporal Fusion Module Finally, we propose the propagative temporal fusion module that fuses the aligned Ref features $\widetilde{h}_t^{Ref}$ with the *temporally aggregated features* $\widehat{h}_t^{\{f,b\}}$ and propagates the fused features $h_t^{\{f,b\}}$ to the next cell (Fig. 5). Note that aligned Ref features $\widetilde{h}_t^{Ref}$ contain Ref features at the current time step, while the temporally aggregated features $\widehat{h}_t^{\{f,b\}}$ contain aggregated temporal Ref features propagated from neighboring recurrent cells. For the successful fusion, the propagative temporal fusion module has to fuse $\widehat{h}_t^{\{f,b\}}$ and $\widetilde{h}_t^{Ref}$ in the way of selecting the Ref features better aligned to the target frame so that well-matched Ref features can keep propagating to the next cell. Otherwise, erroneous Ref features can be accumulated in the pipeline, leading to blurry results.

However, a naïve fusion of the Ref features $\widetilde{h}_t^{Ref}$ is error-prone, as matching is not necessarily accurate. Inspired by [26, 29], we thus perform feature fusion guided by the matching confidences c_t, which guides the fusion module to select only well-matched features in $\widetilde{h}_t^{Ref}$. The fusion module also needs a guidance for propagated Ref features aggregated in $\widehat{h}_t^{\{f,b\}}$. The guidance should accommodate temporal information that coincides with propagated Ref features maintained in the propagation pipeline. To this end, we accumulate matching confidences throughout the propagation pipeline and use the accumulated confidence as the guidance for the temporally aggregated features $\widehat{h}_t^{\{f,b\}}$ during the fusion. Formally, we have:

$$\widetilde{c}_t^{\{f,b\}} = \operatorname{warp}(c_{t\pm1}^{\{f,b\}},\, w_t^{\{f,b\}}), \tag{5}$$

where $c_{t\pm1}^{\{f,b\}}$ is the accumulated matching confidence propagated from neighboring cells. We align the confidence to obtain $\widetilde{c}_t^{\{f,b\}}$ using the optical flow pre-computed in Eq. 3.

For the fusion, we provide matching confidence c_t computed between the current target and reference frames, and we also provide aligned matching confidence $\widetilde{c}_t^{\{f,b\}}$ propagated from neighboring recurrent cells as guidance. The matching confidences are embedded with a convolution layer to consider matching scores of neighboring patches for providing more accurate guidance during the fusion [26]. Formally, the fusion process is defined as:

$$\begin{aligned} h_t^{\{f,b\}} = \{&\operatorname{conv}([c_t,\, \widetilde{c}_t^{\{f,b\}}]) \otimes \\ &\operatorname{conv}([\widetilde{h}_t^{Ref},\, \widehat{h}_t^{\{f,b\}}])\} + \widehat{h}_t^{\{f,b\}}, \end{aligned} \tag{6}$$

where $[,]$ and $\otimes$ indicate concatenation operation and element-wise multiplication, respectively.

For the next cell, we use $\max(,)$ operation to accumulate c_t into $\widetilde{c}_t^{\{f,b\}}$ and pick up a larger confidence score. The accumulation process is defined as:

$$c_t^{\{f,b\}} = \max(c_t,\, \widetilde{c}_t^{\{f,b\}}). \tag{7}$$

Max operation between c_t and $\widetilde{c}_t^{\{f,b\}}$ indirectly imposes the propagative temporal fusion module to selectively fuse and propagate the better matched features between corresponding features, $\widetilde{h}_t^{Ref}$ and $\widehat{h}_t^{\{f,b\}}$, respectively.

4. Training Strategy for Real-World 4× VSR

To train our network, we propose the RealMCVSR dataset, which consists of triplets of ultra-wide, wide-angle, and telephoto videos, where wide-angle and telephoto videos have the same size as ultra-wide videos, but their resolutions are 2× and 4× that of ultra-wide videos. The detail of the dataset is given in Sec. 5. Given video triplets, we train our network to perform 4× super-resolution of an ultra-wide HD video with a wide-angle video as a Ref video for obtaining an 8K video. The resulting 8K video has the same resolution as a telephoto video, but 16× larger in size.

It is worth noting that we use only a wide-angle video as a Ref video. While it may look reasonable to use a telephoto video as an additional Ref video to achieve the resolution of a telephoto video, we found that it does not improve the super-resolution quality much because a telephoto video covers only 1/16 the area of an ultra-wide video. A detailed discussion and experiments are provided in the supplement.

Training our network to produce 8K videos is not trivial as there are no ground-truth 8K videos. While we have wide-angle and telephoto videos, they neither cover the entire area nor perfectly align with an ultra-wide video. To overcome this, we propose a novel training strategy that fully exploits wide-angle and telephoto videos.

Our training strategy consists of pre-training and adaptation stages. In the pre-training stage, we downsample ultra-wide and wide-angle videos 4×. We then train the network to 4× super-resolve a downsampled ultra-wide video using a downsampled wide-angle video as a reference. The training is done in a supervised manner using the original ultra-wide video as the ground-truth. Finally, in the adaptation stage, we fine-tune the network to adapt it to real-world videos of the original sizes. This stage uses a telephoto video as supervision to train the network to recover high-frequency details of a telephoto video. The following subsections describe each stage in more detail.

4.1. Pre-training Stage

In this stage, we train our network using two loss functions: a reconstruction loss motivated by [18, 19, 26] and a multi-Ref fidelity loss. The reconstruction loss minimizes the low- and high-frequency differences between a super-resolved ultra-wide frame I_t^{SR} and the ground-truth ultra-wide frame I_t^{HR}. The reconstruction loss ℓ_{rec} is defined as:

$$\ell_{rec} = ||I^{SR}_{t,blur} - I^{HR}_{t,blur}|| + \lambda_{rec} \sum_i \delta_i(I^{SR}_t, I^{HR}_t), \tag{8}$$

where the subscript *blur* indicates a filtering operation with a 3×3 Gaussian kernel with σ=1.0 and λ_{rec} is a weight for the second term. $\delta_i(X, Y) = \min_j \mathbb{D}(x_i, y_j)$ is the contextual loss [19] that measures the distance between the pixel x_i in X and its most similar pixel y_j in Y under some feature distance measure $\mathbb{D}$, *e.g.*, a perceptual distance [18, 19, 30].

In the first term on the right-hand side in Eq. (8), filtering frames with Gaussian kernels imposes results to follow low-frequency structures of a ground-truth ultra-wide frame I_t^{HR}. The second term enforces the network to follow the high-frequency details of I_t^{HR}. Note that in the second term, we use the contextual loss even for aligned pairs I_t^{SR} and I_t^{HR}, as the loss is verified to be better in boosting the perceptual quality than the perceptual loss [11] designed for aligned pairs [18].

To guide the network to take advantage of multiple Ref frames, we encourage Ref features to keep propagating from one to the next cells. Motivated by [26], we propose a multi-Ref fidelity loss. Given a super-resolved ultra-wide frame I_t^{SR} and ground-truth wide-angle frames $I_{t\in\Omega}^{Ref_{HR}}$, the multi-Ref fidelity loss is defined as:

$$\ell_{Mfid} = \frac{\sum_{t'\in\Omega}\sum_i \delta_i(I^{SR}_t, I^{Ref_{HR}}_{t'}) \cdot c_{t',i}}{\sum_{t'\in\Omega}\sum_i c_{t',i}}, \tag{9}$$

where Ω=$[t-\frac{k-1}{2}, \ldots, t+\frac{k-1}{2}]$ is set of frame indices in a temporal window of size k. We use k=7 in practice. Here, $c_{t',i}$ is the matching confidence used for weighting the distance $\delta_i(I_t^{SR}, I_{t'}^{Ref_{HR}})$. Specifically, during training, pixels of I_t^{SR} with higher matching confidence $c_{t',i}$ are assigned with larger weights for optimization. Eq. 9 enables our network to effectively utilize multiple Ref frames $I_{t\in\Omega}^{Ref}$ and keep the details of multiple Ref frames to flow through the propagation pipeline. Our loss for the pre-training stage is defined as:

$$\ell_{pre} = \ell_{rec}(I^{SR}_t, I^{HR}_t) + \lambda_{pre}\ell_{Mfid}(I^{SR}_t, I^{Ref_{HR}}_{t\in\Omega}). \tag{10}$$

where λ_{pre} is a weight for the multi-Ref fidelity loss for the pre-training stage.

4.2. Adaptation Stage for Real-world 4× VSR

For adaptation, our network takes real-world ultra-wide I_t^{UW} and wide-angle I_t^{Wide} HD frames as LR and Ref frames, respectively. As in the pre-training stage, the adaptation stage separately handles low- and high-frequency of a super-resolved ultra-wide frame I_t^{SR}. However, as there is no ground-truth frame available for I_t^{SR}, we downsample I_t^{SR} and use the input ultra-wide frame I_t^{UW} as the supervision for recovering low-frequency structures. For recovering high-frequency details, we directly utilize telephoto frames $I_{t\in\Omega}^{Tele}$ as the supervision for the proposed multi-Ref fidelity loss ℓ_{Mfid}. The adaptation loss is defined as:

$$\ell_{8K} = ||I^{SR}_{t\downarrow,blur} - I^{UW}_{t,blur}|| + \lambda_{8K}\ell_{Mfid}(I^{SR}_t, I^{Tele}_{t\in\Omega}), \tag{11}$$

where λ_{8K} is a weight for the multi-Ref fidelity loss for the adaptation stage. The first term imposes our network to reconstruct low-frequency structures of input ultra-wide frames, and the second term trains our network to transfer the finest high-frequency details of telephoto frames.

ℓ_{Mfid}	PTF	PSNR↑	SSIM↑	Params (M)
		30.71	0.894	4.2768
✓		31.31	0.913	4.2768
✓	✓	**31.68**	**0.914**	4.2772

Table 1. Quantitative ablation study. The first row corresponds to the baseline model. ℓ_{Mfid} and PTF indicate the models trained with Eq. 9 and propagative temporal fusion module, respectively.

5. Experiments

RealMCVSR Dataset Our RealMCVSR dataset provides real-world HD video triplets concurrently recorded by Apple iPhone 12 Pro Max equipped with triple cameras having fixed focal lengths: ultra-wide (30mm), wide-angle (59mm), and telephoto (147mm). To concurrently record video triplets, we built an iOS app that provides full control over exposure parameters (*i.e*., shutter speed and ISO) of the cameras. For recording each scene, we set the cameras in the auto-exposure mode, where the shutter speeds of the three cameras are synced to avoid varying motion blur across a video triplet. ISOs are adjusted accordingly for each camera to pick up the same exposure. Each video is saved in the MOV format using HEVC/H.265 encoding with the HD resolution (1080 × 1920). The dataset contains triplets of 161 video clips with 23,107 frames in total. The video triplets are split into training, validation, and testing sets, each of which has 137, 8, and 16 triplets with 19,426, 1,141, and 2,540 frames, respectively.

Implementation The network is trained using rectified-Adam [16] with an initial learning rate 2.0×10^{-4}, which is steadily decreased to 1.0×10^{-6} using the cosine annealing strategy [17]. The network is trained for 300k and 50k iterations for the pre-training and adaptation stages, respectively, with $\lambda_{rec} = 0.01$, $\lambda_{pre} = 0.05$, and $\lambda_{8K} = 0.1$. For each iteration, we randomly sample batches of frame triplets from the RealMCVSR training set. For the pre-training stage, we downsample ultra-wide LR and wide-angle Ref frames 4× using bicubic downsampling provided by MATLAB function imresize. We crop patches from each frame in a triplet to have overlapped contents and apply random translation on each crop window. Then, ultra-wide LR frames are cropped to 64 × 64 and 128 × 128 patches for the pre-training and adaptation stages, respectively. Wide-angle and telephoto Ref frames are cropped into patches of 2× and 4× the patch size of LR patches, respectively.

5.1. Ablation Study

To analyze the effect of each component of our model, we conduct ablation studies. First, we validate the effects of the propagative temporal fusion module (Eq. 6) and multi-Ref fidelity loss ℓ_{Mfid} (Eq. 9). To this end, we compare the stripped-out baseline model with its two variants. The baseline model is trained with ℓ_{rec} and ℓ_{Mfid}, but we set the temporal window size k=1 for ℓ_{Mfid}, indicating only a single ground-truth Ref frame is used for computing the loss. Regarding the propagative temporal fusion module, we use a modified one for the baseline model. Specifically, Eq. 6 becomes:

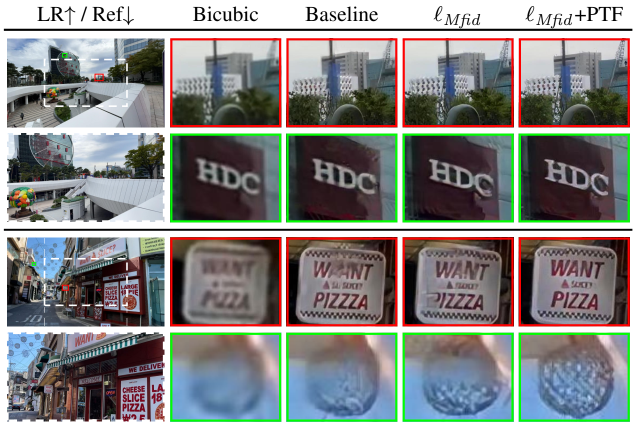

Figure 6. Qualitative ablation study. The first column shows LR and Ref real-world HD inputs. For the rest of the columns, we show zoomed-in cropped 4×SR results of different combinations of modules (Table 1). Red and green boxes indicate inside and outside the overlapped FoV between LR and Ref frames, respectively.

$$h_t^{\{f,b\}} = \{\mathrm{conv}(c_t) \otimes \mathrm{conv}([\tilde{h}_t^{Ref}, \hat{h}_t^{\{f,b\}}])\} + \hat{h}_t^{\{f,b\}}.$$

For the other variants, we recover the key components one by one from the baseline model. For the variant with ℓ_{Mfid}, we train the baseline model with ℓ_{rec} and ℓ_{Mfid} with window size k=7. For the last variant, we attach the propagative temporal fusion module. For quantitative and qualitative comparison, we compare pre-trained models (Sec. 4.1) and their fine-tuned models (Sec. 4.2) on the proposed RealMCVSR test set, respectively.

Table 1 shows quantitative results. The table indicates that compared to the baseline model (the first row in the table), the model trained with ℓ_{Mfid} (the second row) shows much better VSR performance. The model additionally equipped with the propagative temporal fusion module (the third row) achieves the best results in every measure.

Fig. 6 shows a qualitative comparison. As shown in the figure, the model trained with ℓ_{Mfid} (the fourth column of the figure) enhances details inside (red box) and outside (green box) the overlapped FoV much better compared to the results of the baseline model (the third column). The result confirms that ℓ_{Mfid} enforces temporal Ref features to keep streaming through the propagation pipeline to be utilized in reconstructing high-fidelity results. The model attached with the propagative temporal fusion module shows accurately recovered structures and enhanced details for both inside and outside the overlapped FoV (the last column). This demonstrates the propagative temporal fusion module promotes well-matched Ref features to be fused and to flow through the propagation pipeline.

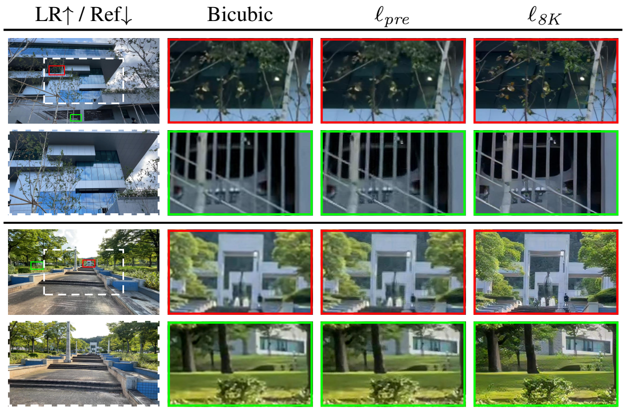

Figure 7. Ablation study on the two-stage training strategy.

We also validate the effects of the proposed training strategy. Specifically, we qualitatively compare between the model pre-trained with the pre-training loss ℓ_{pre} (Eq. 10) and the model fine-tuned with the adaptation loss ℓ_{8K} (Eq. 11). For comparison, we show 8K VSR results given real-world HD videos. Note that in the real-world scenario, there is no ground-truth available for a quantitative comparison. Fig. 7 shows the results. The pre-trained model does not improve details of a real-world input, due to the domain gap between real-world inputs and downsampled inputs (the third column). However, the fine-tuned model shows much higher fidelity results compared to the pre-trained model (the last column), thanks to the adaptation stage that trains the network to well adapt to real-world videos.

5.2. Comparison on RealMCVSR Dataset

In this section, we compare our method with previous state-of-the-art approaches: SRCNN [6], RCAN [31], TTSR [29], DCSR [26], EDVR [27], BasicVSR [4], and IconVSR [4]. SRCNN and RCAN are SISR models that take only a single LR frame. TTSR and DCSR are RefSR models fed with a pair of LR and Ref frames. EDVR is a sliding window-based VSR model that takes multiple frames in a local temporal window. BasicVSR and IconVSR are VSR models with a bidirectional recurrent framework, where each video frame is fed to a recurrent cell for each time step. We train each model with the proposed RealMCVSR dataset and the code provided by the authors.

Quantitative Comparison Table 2 shows a quantitative comparison, where ultra-wide HD frames and their 4× downsampled ones are used as ground-truths and inputs, respectively. For comparison, we use our model pre-trained with Eq. 10 (Ours). Moreover, to consider the trade-off between the model size and SR quality, we show the results of a smaller model with fewer parameters (Ours-*small*) and a larger model (Ours-IR) attached with information-refill and coupled propagation modules proposed in [4]. We also compare our models trained only with the ℓ_1 loss function (models indicated with -ℓ_1), for a fair comparison with the previous models trained with pixel-based losses, such as ℓ_1, ℓ_2, and ℓ_{ch} (Charbonnier loss [12]), which are known for having an advantage in PSNR over perceptual-based loss [11].

	Model	PSNR↑	SSIM↑	Params (M)
SISR	Bicubic	26.65	0.800	-
	SRGAN [13]	29.38	0.877	0.734
	RCAN-ℓ_1 [31]	31.07	0.915	15.89
RefSR	TTSR [29]	30.31	0.905	6.730
	TTSR-ℓ_1 [29]	30.83	0.911	6.730
	DCSR [26]	30.63	0.895	5.419
	DCSR-ℓ_1 [26]	32.43	0.933	5.419
VSR	EDVR-M-ℓ_{ch} [27]	33.26	0.946	3.317
	EDVR-ℓ_{ch} [27]	33.47	0.948	20.63
	BasicVSR-ℓ_{ch} [4]	33.66	0.951	4.851
	IconVSR-ℓ_{ch} [4]	33.80	0.951	7.255
RefVSR	Ours-*small*	31.63	0.912	1.052
	Ours-*small*-ℓ_1	33.88	0.951	1.052
	Ours	31.68	0.914	4.277
	Ours-ℓ_1	34.74	0.958	4.277
	Ours-IR	31.73	0.916	4.774
	Ours-IR-ℓ_1	**34.86**	**0.959**	4.774

Table 2. Quantitative evaluation on the RealMCVSR test set.

In Table 2, while RefSR methods show a better performance than SISR methods, our methods outperform all previous ones. Interestingly, VSR methods outperform RefSR methods that are additionally fed with Ref frames. However, this is not particularly true if we measure the performance on the regions of the SR frame corresponding to different FoV ranges. Table 3 shows the results. For comparison, we measure the SR quality for the region inside the overlapped FoV (0%–50%) between an ultra-wide SR and a wide-angle Ref frames. For outside the overlapped FoV, we measure SR performance for the banded regions at different FoV ranges from the overlapped FoV (50%) to full FoV (100%). In the table, DCSR [26] outperforms IconVSR [4] for the overlapped FoV (0%–50%) between an input and Ref frames, while IconVSR outruns DCSR for the rest of the regions. Our models exceed all models for all regions.

Note that in Table 3, our models show a performance gap between regions inside (0%–50%) and outside (50%–100%) the overlapped FoV. However, compared to the PSNR/SSIM gap of DCSR (8.5% / 4.2%), our models show much smaller gap (Ours-ℓ_1 : 4.2% / 1.8% and Ours-IR-ℓ_1 : 4.2% / 1.6%). The result implies the proposed architecture effectively utilizes neighboring Ref features for recovering regions both inside and outside of the overlapped FoV.

Qualitative Comparison For the qualitative comparison, we show 8K (4320 × 7280) 4×SR video results given real-world HD (1080 × 1920) videos. For the comparison, we select the best models from each SISR, RefSR, and VSR approaches: RCAN [31], DCSR [26], and IconVSR [4], respectively, according to their quantitative performance shown with the RealMCVSR test set. We train each model with the proposed training strategy (Sec. 4).

	Model	PSNR / SSIM measured for regions in the indicated FoV range						Params
		0%–50%	50%–60%	50%–70%	50%–80%	50%–90%	50%–100%	(M)
SISR	Bicubic	25.38 / 0.757	26.30 / 0.785	26.42 / 0.789	26.71 / 0.798	26.99 / 0.801	27.29 / 0.815	-
	RCAN-ℓ_1 [31]	29.77 / 0.895	30.69 / 0.908	30.86 / 0.910	31.17 / 0.914	31.50 / 0.918	31.80 / 0.921	15.89
RefSR	DCSR-ℓ_1 [26]	34.90 / 0.963	31.96 / 0.927	31.61 / 0.921	31.58 / 0.919	31.81 / 0.921	31.93 / 0.923	5.419
VSR	IconVSR-ℓ_{ch} [4]	32.79 / 0.946	33.43 / 0.949	33.60 / 0.950	33.89 / 0.951	34.19 / 0.953	34.40 / 0.953	7.255
RefVSR	Ours-ℓ_1	36.02 / 0.971	34.59 / 0.958	34.31 / 0.956	34.23 / 0.954	34.40 / 0.955	34.50 / 0.954	4.277
	Ours-IR-ℓ_1	**36.14 / 0.971**	**34.66 / 0.959**	**34.40 / 0.956**	**34.34 / 0.955**	**34.52 / 0.955**	**34.63 / 0.955**	4.774

Table 3. Quantitative results measured with varying FoV range. The center 50% of FoV in an ultra-wide SR frame is overlapped with the FoV of a wide-angle reference frame. Here, 0%–50% indicates the region inside the overlapped FoV, and 50%–100% is the region outside the overlapped FoV. 50%–60% means the banded region between the center 50% and 60% of an ultra-wide SR frame.

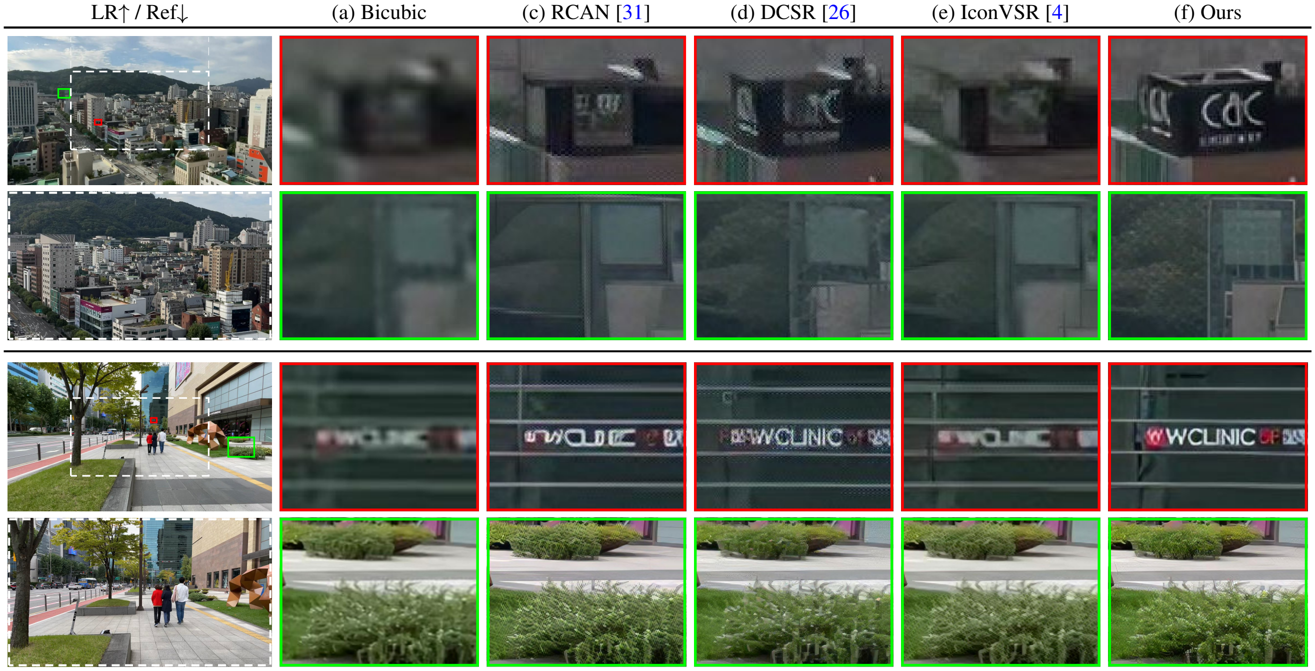

Figure 8. Qualitative comparison on 8K 4×SR video results from real-world HD videos.

Fig. 8 shows a qualitative comparison for 8K 4×SR results from real-world HD videos. The results show that non-reference-based SR methods, RCAN and IconVSR, tend to over-exaggerate textures, while non-textured regions tend to be overly smoothed out. The RefSR method, DCSR, shows better fidelity than RCAN and IconVSR in the overlapped FoV (red box). However, DCSR tends to smooth out regions outside the overlapped FoV (green box). Our method shows the best result compared to the previous ones. Compared to DCSR, our model robustly reconstructs finer details with balanced fidelity between regions inside and outside the overlapped FoV. Moreover, the details and textures reconstructed outside the FoV are more photo-realistic.

6. Conclusion

We proposed the first RefVSR framework with the practical focus on videos captured in an asymmetric multi-camera setting. To efficiently utilize a Ref video sequence, we adopted a bidirectional recurrent framework and proposed the propagative temporal fusion module to fuse and propagate Ref features well-matched to LR features. To train and validate the network, we provided the RealMCVSR dataset consisting of real-world HD video triplets. An adaptation training strategy is proposed to fully utilize video triplets in the dataset. In the experiments, we verified the effects of key components in our model, and our model achieves the state-of-the-art 4×VSR performance.

Limitation As previous RefSR methods [26, 28, 29, 32], our network consumes quite an amount of memory for applying global matching between real-world HD frames. We plan to develop a memory-efficient RefVSR framework.

Acknowledgments We thank Hyeongseok Son for helpful discussions and Jihye Kim and Anna Choi for their help in collecting the RealMCVSR dataset. This work was supported by the Ministry of Science and ICT, Korea, through IITP grants (SW Star Lab, 2015-0-00174; AI Innovation Hub, 2021-0-02068; Artificial Intelligence Graduate School Program (POSTECH), 2019-0-01906) and NRF grants (2018R1A5A1060031; 2020R1C1C1014863).

References

[1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (TOG)*, 28(3), 2009. 2

[2] Vivek Boominathan, Kaushik Mitra, and Ashok Veeraraghavan. Improving resolution and depth-of-field of light field cameras using a hybrid imaging system. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, 2014. 1, 2

[3] Jose Caballero, Christian Ledig, Andrew Aitken, Alejandro Acosta, Johannes Totz, Zehan Wang, and Wenzhe Shi. Real-time video super-resolution with spatio-temporal networks and motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[4] Kelvin CK Chan, Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7, 8

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 4

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2014. 7

[7] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 2

[8] Yan Huang, Wei Wang, and Liang Wang. Video super-resolution via bidirectional recurrent convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 40(4):1015–1028, 2018. 2

[9] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2020. 2

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and koray kavukcuoglu. Spatial transformer networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 4

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2016. 5, 7

[12] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[13] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 7

[14] Sheng Li, Fengxiang He, Bo Du, Lefei Zhang, Yonghao Xu, and Dacheng Tao. Fast spatio-temporal residual network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[15] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2020. 2

[16] Liyuan Liu, Haoming Jiang, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 6

[17] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with restarts. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017. 6

[18] Roey Mechrez, Itamar Talmi, Firas Shama, and Lihi Zelnik-Manor. Maintaining natural image statistics with the contextual loss. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, 2018. 5

[19] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2018. 5

[20] Anurag Ranjan and Michael Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[21] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[22] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3

[23] Gyumin Shim, Jinsun Park, and In So Kweon. Robust reference-based super-resolution with similarity-aware deformable convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 4

[25] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[26] Tengfei Wang, Jiaxin Xie, Wenxiu Sun, Qiong Yan, and Qifeng Chen. Dual-camera super-resolution with aligned attention modules. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 4, 5, 7, 8

[27] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019. 2, 7

[28] Yanchun Xie, Jimin Xiao, Mingjie Sun, Chao Yao, and Kaizhu Huang. Feature representation matters: End-to-end learning for reference-based image super-resolution. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 4, 8

[29] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4, 7, 8

[30] Xuaner Zhang, Qifeng Chen, Ren Ng, and Vladlen Koltun. Zoom to learn, learn to zoom. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[31] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2018. 7, 8

[32] Zhifei Zhang, Zhaowen Wang, Zhe L. Lin, and Hairong Qi. Image super-resolution by neural texture transfer. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 8

[33] Yaping Zhao, Mengqi Ji, Ruqi Huang, Bin Wang, and Shengjin Wang. Efenet: Reference-based video super-resolution with enhanced flow estimation. *arXiv preprint*, arXiv:2110.07797, 2021. 2

[34] Haitian Zheng, Mengqi Ji, Lei Han, Ziwei Xu, Haoqian Wang, Yebin Liu, and Lu Fang. Learning cross-scale correspondence and patch-based synthesis for reference-based super-resolution. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2017. 1, 2

[35] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In *Proceedings of the Springer European Conference on Computer Vision (ECCV)*, 2018. 1, 2