Revisiting Temporal Alignment for Video Restoration

Kun Zhou[1,2*] Wenbo Li[3*] Liying Lu[3] Xiaoguang Han[1] Jiangbo Lu[2†]
[1]The Chinese University of Hong Kong (Shenzhen), [2]SmartMore Corporation
[3]The Chinese University of Hong Kong
kunzhou@link.cuhk.edu.cn, {wenboli,lylu}@cse.cuhk.edu.hk
hanxiaoguang@cuhk.edu.cn,jiangbo.lu@gmail.com

Abstract

Long-range temporal alignment is critical yet challenging for video restoration tasks. Recently, some works attempt to divide the long-range alignment into several sub-alignments and handle them progressively. Although this operation is helpful in modeling distant correspondences, error accumulation is inevitable due to the propagation mechanism. In this work, we present a novel, generic iterative alignment module which employs a gradual refinement scheme for sub-alignments, yielding more accurate motion compensation. To further enhance the alignment accuracy and temporal consistency, we develop a non-parametric reweighting method, where the importance of each neighboring frame is adaptively evaluated in a spatial-wise way for aggregation. By virtue of the proposed strategies, our model achieves state-of-the-art performance on multiple benchmarks across a range of video restoration tasks including video super-resolution, denoising and deblurring.

1. Introduction

Frame alignment plays an essential role in aggregating temporal information in video restoration tasks, e.g., video super-resolution (Video SR), video deblurring, and video denoising. In recent years, great attempts have been made to study this problem. Especially, deep learning-based methods are successful in building temporal correspondences and achieve promising results.

The existing alignment methods can be roughly categorized into two classes: (i) *independent alignment* that conducts frame-to-frame alignments totally independently (see Fig. 2(a)) and (ii) *progressive alignment* that performs temporally consecutive alignments sequentially in a recursive manner (see Fig. 2(b)). Those independent alignment approaches typically focus on designing effective feature descriptors and motion estimation modules to improve the performance. For example, EDVR [29] develops pyramid, cascading and deformable convolutions (PCD) for more accurate alignment. Whereas, without exploiting the correlations between multiple alignments, this strategy is still facing challenges to estimate the long-range motion fields. The second line typically adopts a recurrent framework for gradual alignment. Taking BasicVSR [3] for example, the authors propose an optical-flow-based recurrent architecture for video super-resolution. They predict the bidirectional optical flow between two neighboring frames and then conduct a bidirectional propagation, where the temporal information is aggregated by warping image features produced by previous steps. This kind of methods is mainly proposed to model long-range dependencies since it only needs to handle relatively small motion between neighboring frames in one step. However, such chain-rule-based propagation has no chance to correct the misalignment caused by previous steps and may suffer from the error accumulation issue.

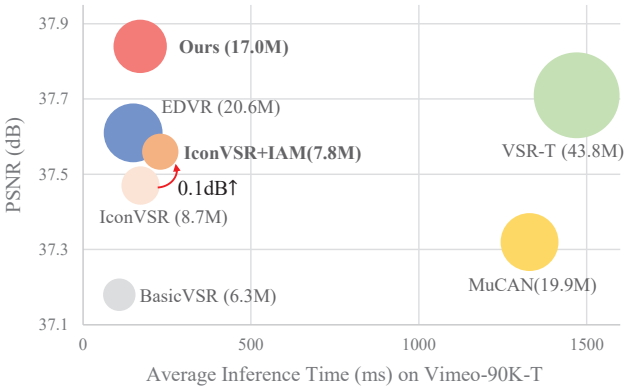


Figure 1. Performance and efficiency comparison on Vimeo-90K-T [33]. Besides high PSNR and fast inference, our alignment algorithm can be easily integrated into existing frameworks (e.g., IconVSR [3]) to further improve performance. Circle sizes are set proportional to the numbers of parameters.

As illustrated in Fig. 2(c), we observe that different long-range alignments (A_i) actually share some *sub-alignments* (a_i), e.g., a_1 is shared among A_1, A_2 and A_3, so as a_2 in A_2 and A_3. How can we utilize this property

*Equal contribution
†Corresponding author

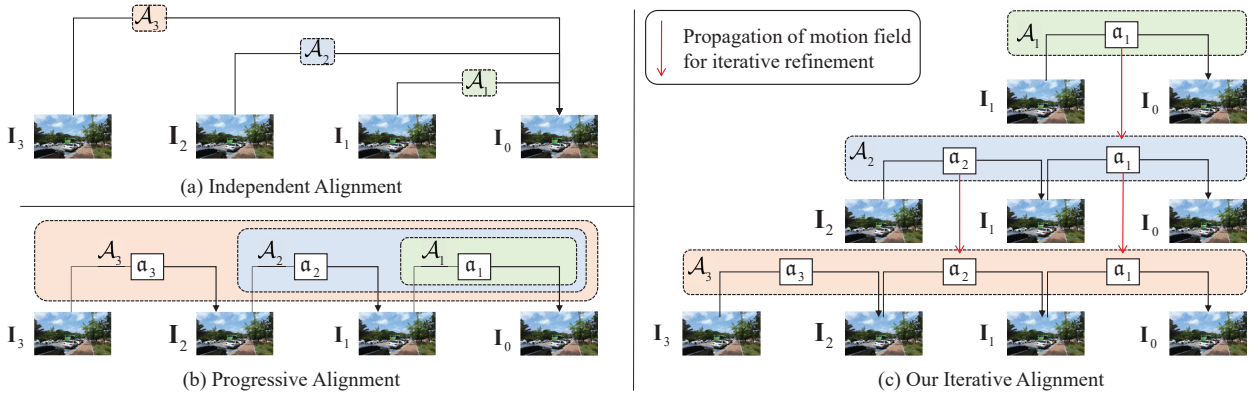


Figure 2. Three alignment strategies in video restoration tasks. (a) Independent alignment that estimates frame-to-frame correspondences in isolation. (b) Progressive alignment that performs multiple alignments sequentially. (c) Our proposed iterative alignment scheme that performs gradual refinement for shared sub-alignments. $\mathcal{A}_k$ refers to the k-th temporal alignment and a_i is the i-th sub-alignment.

to improve the accuracy of the shared sub-alignments? In this work, we propose an *iterative alignment* module (IAM) built upon the progressive alignment strategy to gradually refine the shared sub-alignments. For a specific shared sub-alignment (e.g., a_2 in A_2 and A_3), the previously estimated result (a_2 in A_2) is used as a prior in the current iteration (a_2 in A_3). Our IAM has two merits over the progressive alignment scheme. First, the progressive alignment only conducts a single prediction for each sub-alignment so that misalignment can not be corrected. In contrast, our IAM refines each sub-alignment iteratively, yielding more accurate alignment. Second, the progressive alignment performs multi-frame aggregation based on a chain-like propagation so that misalignment will be propagated to the end. In our IAM, each neighboring frame is aligned through individual propagation, making it more reliable. Furthermore, to reduce the computational complexity, we elaborate a simple yet efficient alignment unit for temporal sub-alignments.

From Fig. 1, it is observed that our alignment algorithm yields high inference efficiency and superior performance compared with state-of-the-art video SR methods. Particularly, our IAM can be easily plugged into existing deep models. For example, by replacing the original independent alignment module of IconVSR [3] with our "IAM" (denoted as "IconVSR+IAM" in Fig. 1), the PSNR is boosted from 37.47dB to 37.56dB on Vimeo-90K-T [33], while reducing the number of parameters from 8.7M to 7.8M.

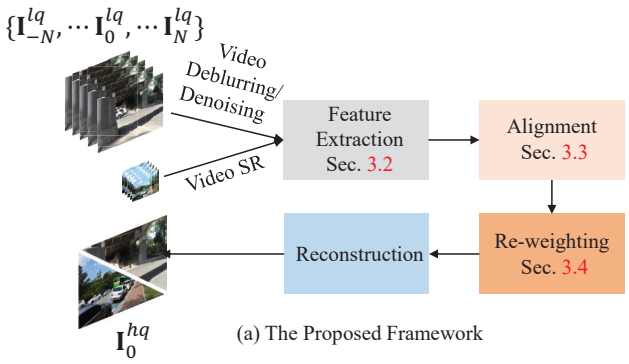
Besides, the aggregation of multiple aligned frames remains an essential step, for the purpose of preserving details while eliminating alignment errors. Modern restoration systems either employ a sequence of convolutions to directly fuse the aligned features [3, 28] or adopt spatial-temporal adaptive aggregation strategies [9, 13, 15, 16, 19, 29, 31]. However, all these methods solely rely on the learned parameters, raising the risk of overfitting on a specific domain. In this work, we propose a non-parametric re-weighting module, where two strategies are designed to explicitly evaluate the spatially-adaptive importance of different frames. First, we explore the accuracy of alignments. Patches in the aligned frames are compared with the counterparts in the reference frame, and those of high similarity are assigned with larger weights during fusion. Second, to evaluate the consistency of alignments, we compute the pixel-wise L2 distances of the aligned frames with their average. Pixels with smaller distances are considered to be more consistent with other frames and hence are assigned with larger weights. The proposed re-weighting module is parameterless and hence can be plugged into other models.

The main contributions are summarized as:

- We rethink issues of the progressive alignment and accordingly propose an iterative alignment scheme, yielding more accurate estimation, especially over long-range correspondences.
- We propose a non-parametric re-weighting module that simultaneously evaluates the alignment accuracy and temporal consistency.
- The quantitative and qualitative results justify the state-of-the-art performance of our method across several video restoration tasks.

2. Related Work

Temporal Alignment. Many video restoration approaches [3,22,30,33] perform independent temporal alignment between neighboring frames with the central frame. Various strategies have been proposed to improve the performance. For example, to fill the domain gap between optical flow estimation and video SR tasks, TOF [33] integrates a task-oriented flow module into their VSR framework for end-to-end training. Pan *et al*. [22] develop CNNs to estimate the optical flow and the latent frame simultaneously.



(a) The Proposed Framework

Figure 3. A general framework for video restoration tasks. There are four components including a frame feature extraction module, an iterative alignment module, an adaptive re-weighting module and a reconstruction module.

Later on, some methods start to develop adaptive kernel-based schemes [14,24,28,29,32,34,35] to perform the alignment and process the occlusion simultaneously. EDVR [29] proposes a coarse-to-fine alignment algorithm to tackle the large displacement. However, these independent alignment models only focus on exploring correlations between two frames in isolation. It is still challenging to handle long-range alignments.

Another line of work [3, 4] begins to explore a progressive alignment strategy for video restoration tasks. To alleviate the challenges of long-range alignment, they typically split multiple long-range alignments into several sub-alignments. Those sub-alignments are subsequently processed progressively. In BasicVSR [3], a pre-trained SPyNet [23] is utilized to estimate motion fields of each sub-alignment between adjacent frames. Then, they progressively aggregate the temporal information by warping image features produced by previous steps. The progressive alignment scheme makes it effective in handling long-range alignment. Based on BasicVSR, BasicVSR++ [4] presents a second-order propagation and motion field residual learning method to improve the accuracy of sub-alignments. However, inaccurately estimated motion fields of some sub-alignments will wrongly warp the image features. The misaligned information is subsequently propagated and aggregated in the later steps, resulting in error accumulation. In this work, we propose an iterative alignment algorithm built upon the progressive alignment scheme. Each sub-alignment is estimated and refined gradually, largely improving the accuracy of alignment.

Feature Fusion. The majority of video restoration methods fuse the aligned frames for temporal information aggregation by feature concatenation followed by a convolution [18, 28, 33]. For example, FastDVD [26] divides consecutive frames into different groups and designs a two-stage convolutional neural network for multi-frame fusion.

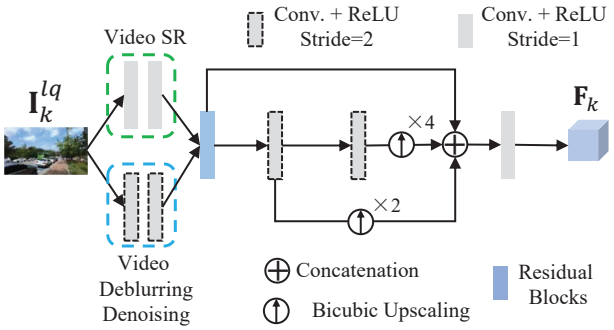


Figure 4. Overview of our feature extraction module.

In addition, more effective aggregation strategies have been proposed by applying spatial or temporal attention-based mechanism [9, 13, 19, 31]. Isobe *et al*. [9] design a frame-rate-aware group attention, which can handle various levels of motions. In [30], a motion robustness analysis is adopted to fuse temporal information, where different confidence scores are assigned to the local neighbors of each pixel for merging. Inspired by this work, we design an adaptive re-weighting module for information aggregation, considering both the accuracy and consistency of the alignment.

3. Methodology

3.1. Overview

Figure 3 shows the proposed framework. Our goal is to reconstruct a high-quality image $\mathbf{I}_0^{hq}$ from $2N+1$ consecutive low-quality images $\{\mathbf{I}_{-N}^{lq}, \cdots, \mathbf{I}_0^{lq}, \cdots, \mathbf{I}_N^{lq}\}$. In the feature extraction module, the input frames are first downsampled with strided convolutions for video deblurring/denoising, while being processed under the same resolution for video SR. Then we utilize the proposed IAM to align input frames referring to the central frame. For simplicity, we only consider the one-side alignment in the following as the other side is processed symmetrically. Afterwards, an adaptive re-weighting module is designed to fuse the aligned features. Finally, the $\mathbf{I}_0^{hq}$ is obtained by adding the predicted residue to the original (for video deblurring/denoising) or upsampled (for video SR) input image.

3.2. Feature Extraction

As illustrated in Fig. 4, we conduct feature extraction to transform a RGB frame $\mathbf{I}_k^{lq}$ to high-dimensional feature maps $\mathbf{F}_k$. We first utilize two convolutions with strides of 2 to downsample the feature resolutions for video deblurring and denoising (highlighted in blue dotted box in Fig. 4) for computational efficiency, while keeping the same resolution for video SR (highlighted in green dotted box in Fig. 4). Then we utilize another two convolutions with stride of 2 to obtain the pyramid representations of the input frames. At last, we fuse the pyramid features with a single convolution.

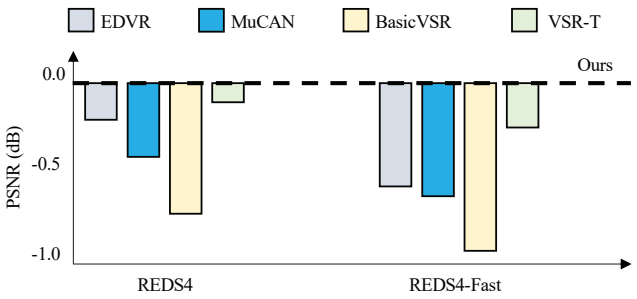


Figure 5. PSNR differences of four SOTA video SR methods [2, 3, 14, 29] compared to our method (dotted line) on REDS4 [21] and REDS4-Fast. The smaller the value, the larger the gap.

3.3. Temporal Alignment

Temporal alignment aims to align multiple neighboring features $\{\mathbf{F}_{-N}, \cdots, \mathbf{F}_{-1}, \mathbf{F}_1, \cdots, \mathbf{F}_N\}$ to a reference $\mathbf{F}_0$. Let $\mathcal{A}_k$ be the k-th temporal alignment between the neighboring frame $\mathbf{F}_k$ and the reference frame $\mathbf{F}_0$, then we have

$$\mathcal{A}_k(\mathbf{F}_k, \mathbf{F}_0) = \hat{\mathbf{F}}_k^0, \ k \in \{-N, \cdots, -1, 1, \cdots, N\}, \quad (1)$$

where $\hat{\mathbf{F}}_k^0$ is the aligned result.

3.3.1 Progressive Alignment

In order to facilitate the long-range alignment, some recent methods [4] adopt a progressive alignment strategy. For the alignment $\mathcal{A}_k$, they divide it into sequential sub-alignments $\{a_k, a_{k-1}, \cdots, a_1\}$ to gradually align the feature $\mathbf{F}_k$ to the reference frame $\mathbf{F}_0$. We use a_i to represent the sub-alginment from $\mathbf{F}_i$ to $\mathbf{F}_{i-1}$:

$$a_i : \mathbf{F}_i \rightarrow \mathbf{F}_{i-1}. \quad (2)$$

As illustrated in Fig. 2(b), all neighboring frames are processed through the chained sub-alignments, indicating that the latter sub-alignments strongly depend on the former predictions. Consequently, the error incurred by an intermediate inaccurate sub-alignment will be propagated and accumulated till the end, leading to inferior performance. To alleviate the issue of error accumulation and boost the restoration quality, we propose an iterative alignment algorithm to focus on improving the accuracy of each sub-alignment a_i.

3.3.2 Iterative Alignment

Unlike the progressive alignment that conducts each sub-alignment only once, our algorithm iteratively refines the sub-alignments based on the previous estimation. As shown in Fig. 2(c), we start from the alignment $\mathcal{A}_1$, which only contains the sub-alignment $a_1 : \mathbf{F}_1 \rightarrow \mathbf{F}_0$, described as:

$$\mathcal{A}_1 : a_1(\mathbf{F}_1, \mathbf{F}_0, t = 1) \Rightarrow \hat{\mathbf{F}}_1^0, \mathbf{h}_1^1, \quad (3)$$

where $\hat{\mathbf{F}}_k^{i-1}$ refers to the aligned result of sub-alignment a_i in $\mathcal{A}_k$. In Eq. 3, $\mathbf{h}_i^t$ represents the estimated motion field of the sub-alignment a_i after being refined t times.

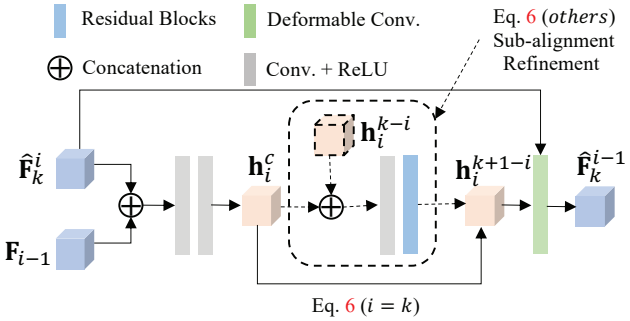


Figure 6. Illustration of our iterative sub-alignment unit for the a_i of $\mathcal{A}_k$. $\hat{\mathbf{F}}_k^i$ is the source feature and $\mathbf{F}_{i-1}$ is the target feature. The iterative refinement is highlighted in dashed box. $\hat{\mathbf{F}}_k^{i-1}$ is the aligned result and $\mathbf{h}_i^{k+1-i}$ is the refined motion field of a_i.

After that, we consider the next alignment $\mathcal{A}_2$ by sequentially performing two sub-alignments $\{a_2, a_1\}$:

$$\mathcal{A}_2 : \begin{cases} a_2(\mathbf{F}_2, \mathbf{F}_1, t = 1) \Rightarrow \hat{\mathbf{F}}_2^1, \mathbf{h}_2^1, \\ a_1(\hat{\mathbf{F}}_2^1, \mathbf{F}_0, \mathbf{h}_1^1, t = 2) \Rightarrow \hat{\mathbf{F}}_2^0, \mathbf{h}_1^2. \end{cases} \quad (4)$$

For a_1 in $\mathcal{A}_2$, it has already been carried out in $\mathcal{A}_1$ once. Thus we fuse the pre-estimated motion field $\mathbf{h}_1^1$ of a_1 in $\mathcal{A}_1$ to refine a_1 in $\mathcal{A}_2$, formulated as an iterative optimization.

For the subsequent alignment $\mathcal{A}_3$, two sub-alignments $\{a_2, a_1\}$ will be refined as:

$$\mathcal{A}_3 : \begin{cases} a_3(\mathbf{F}_3, \mathbf{F}_2, t = 1) \Rightarrow \hat{\mathbf{F}}_3^2, \mathbf{h}_3^1, \\ a_2(\hat{\mathbf{F}}_3^2, \mathbf{F}_1, \mathbf{h}_2^1, t = 2) \Rightarrow \hat{\mathbf{F}}_3^1, \mathbf{h}_2^2, \\ a_1(\hat{\mathbf{F}}_3^1, \mathbf{F}_0, \mathbf{h}_1^2, t = 3) \Rightarrow \hat{\mathbf{F}}_3^0, \mathbf{h}_1^3. \end{cases} \quad (5)$$

It can be concluded that, apart from the first sub-alignment a_k in $\mathcal{A}_k$, all other sub-alignments are optimized at least twice. There are two merits: (i) The sub-alignments will be more accurate through our iterative refinements. (ii) The sub-alignments not only rely on the pre-aligned features but also the pre-estimated motion field, making it more reliable.

To verify our claim, we evaluate our algorithm together with recent video SR models [2, 3, 14, 29] on REDS4 [21] and REDS4-Fast [1]. As shown in Fig. 5, our model achieves the best performance over the competing methods. Particularly, our method brings about significant improvement in the context of large motion, demonstrating the effectiveness of our IAM in the long-range alignment.

3.3.3 Sub-alignment Unit

In Sec. 3.3.2, we describe the iterative alignment algorithm in detail. It is observed that for $2N$ neighboring frames, our method requires $N(N+1)$ sub-alignments. In contrast, the independent and progressive alignment schemes

[1] REDS4-Fast is a subset of REDS4 [21] with an average motion magnitude of **9.4** pixels, much larger than the average of REDS4 of **4.3** pixels. The optical flows are calculated by RAFT [27].

only need $2N$ (sub-)alignments. So it is critical to design a simple sub-alignment unit for computational efficiency. To this end, two improvements have been proposed. (i) While the previous methods [3, 29] typically adopt a pyramid alignment scheme that performs multiple-scale processing in the alignment phase, we adopt an early multi-scale fusion strategy in the feature extraction phase so that our IAM only performs single-scale alignments. (ii) We develop a lightweight sub-alignment unit with much fewer parameters than other methods [2, 29]. Specifically, we use a compact structure of residual blocks to reduce computational overhead (see details in supplementary materials.).

Fig. 6 shows the structure of our sub-alignment unit. Taking the i-th sub-alignment a_i of $\mathcal{A}_k$ for example, we first utilize two convolutions followed by ReLU activation to estimate the initialized motion field $\mathbf{h}_i^c$ from the concatenation of source feature $\hat{\mathbf{F}}_k^i$ and target feature $\mathbf{F}_{i-1}$. After that, there are two cases for the prediction $\mathbf{h}_i^{k+1-i}$ of a_i:

$$\mathbf{h}_i^{k+1-i} = \begin{cases} \mathbf{h}_i^c, & i = k\,, \\ \theta(\mathbf{h}_i^c, \mathbf{h}_i^{k-i}), & others\,. \end{cases} \tag{6}$$

If a_i is the first sub-alignment of $\mathcal{A}_k$ $(i = k)$, then no historical prediction can be reused to refine a_i. As a result, we simply set $\mathbf{h}_i^c$ as the estimated motion field of a_i. Otherwise, we will take the last estimation $\mathbf{h}_i^{k-i}$ together with the current estimation $\mathbf{h}_i^c$ as input and utilize a single convolution followed by two residual blocks (dubbed as θ) to refine the prediction. Finally, we adopt a deformable convolution [6] to adaptively sample contents from the source feature $\hat{\mathbf{F}}_k^i$:

$$\hat{\mathbf{F}}_k^{i-1} = \mathrm{DConv}(\hat{\mathbf{F}}_k^i, \mathbf{F}_{i-1}, \mathbf{h}_i^{k+1-i}) \tag{7}$$

Specially, if a_i is the first sub-alignment in $\mathcal{A}^k$ $(i = k)$, Eq. 7 can be written as:

$$\hat{\mathbf{F}}_k^{k-1} = \mathrm{DConv}(\mathbf{F}_k, \mathbf{F}_{k-1}, \mathbf{h}_k^1) \tag{8}$$

The sub-alignment unit is shared for all sub-alignments, largely reducing the number of learnable parameters.

3.4. Adaptive Re-weighting

Although the temporal alignment module performs motion compensation for neighboring frames, it remains vital to fuse them in an effective way. Recently, convolution-based attention mechanism becomes popular to aggregate multi-frame information [9, 13, 19, 31]. By contrast, we present a non-parametric re-weighting module to explicitly evaluate the spatially-adaptive importance of aligned frames from two perspectives. First, we evaluate the accuracy of aligned frames with respect to the reference frame. Second, we measure the consistency of aligned neighboring frames. Fig. 7 describes the pipeline of our re-weighting module.

Accuracy-Based Re-weighting. As shown in Fig. 7(a), we measure the accuracy of aligned frames. For the reference frame $\mathbf{F}_0$, the feature vector at position (x, y) is denoted as $\mathbf{v}_0$, i.e., $\mathbf{v}_0 = \mathbf{F}_0(x, y)$. We find its corresponding

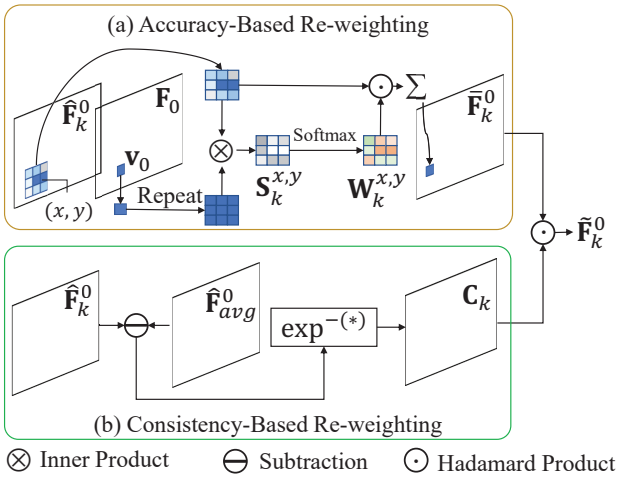


Figure 7. Adaptive re-weighting module. There are two branches: (a) the accuracy-based re-weighting branch for measuring the accuracy of alignment, (b) the consistency-based re-weighting branch for evaluating the consistency of the aligned frames.

3×3 patch centered at the same position in the k-th aligned frame $\hat{\mathbf{F}}_k^0$. For each feature vector on this patch, we calculate its cosine similarity (normalized inner product) with respect to $\mathbf{v}_0$ as:

$$\mathbf{S}_k^{x,y}(\Delta x, \Delta y) = \frac{\hat{\mathbf{F}}_k^0(x + \Delta x, y + \Delta y)}{\left\|\hat{\mathbf{F}}_k^0(x + \Delta x, y + \Delta y)\right\|_2} \otimes \frac{\mathbf{v}_0}{\|\mathbf{v}_0\|_2}\,, \tag{9}$$

where $\mathbf{S}_k^{x,y}$ is the 3×3 similarity map at position (x, y) and $\otimes$ represents the inner product. $(x + \Delta x, y + \Delta y)$ is the coordinate of feature vector where $\Delta x, \Delta y \in \{-1, 0, 1\}$. Then a Softmax function is applied to $\mathbf{S}_k^{x,y}$ in the spatial dimension, yielding the pixel-wise weights as:

$$\mathbf{W}_k^{x,y} = \mathrm{Softmax}(\mathbf{S}_k^{x,y})\,. \tag{10}$$

Then $\mathbf{W}_k^{x,y}$ is used to fuse feature vectors on the 3×3 patch, and the re-weighted result $\bar{\mathbf{F}}_k^0(x, y)$ is obtained as:

$$\bar{\mathbf{F}}_k^0(x, y) = \sum_{\Delta x, \Delta y} \mathbf{W}_k^{x,y}(\Delta x, \Delta y) \odot \hat{\mathbf{F}}_k^0(x + \Delta x, y + \Delta y)\,. \tag{11}$$

where $\odot$ denotes the Hadamard product.

Consistency-Based Re-weighting. We first calculate the average of aligned neighboring frames yielding $\hat{\mathbf{F}}_{avg}^0$, as illustrated in Fig. 7(b). For the k-th aligned frame $\hat{\mathbf{F}}_k^0$, we evaluate its consistency with other aligned frames as

$$\mathbf{C}_k = \exp(\alpha \cdot \|\hat{\mathbf{F}}_k^0 - \hat{\mathbf{F}}_{avg}^0\|_2^2)\,, \tag{12}$$

where α is set to -1 in our experiments. It is noted that $\mathbf{C}_k$ maintains the same shape as $\hat{\mathbf{F}}_k^0$.

Task	Video SR	Video Deblurring	Video Denoising
Configuration	$M(128), B(40)$	$M(128), B(10)$ $M(128), B(40)$	$M(64), B(10)$
GPUs	6	6	2
Patch Reso.	64×64	128×128	128×128
nFrames	5(7)	5	5
Mini-Batch	4	4	16

Table 1. The training and network configurations.

Finally, we multiply the accuracy-based re-weighted feature $\bar{\mathbf{F}}_k^0$ to the consistency map $\mathbf{C}_k$ and obtain the result:

$$\tilde{\mathbf{F}}_k^0 = \bar{\mathbf{F}}_k^0 \odot \mathbf{C}_k \,. \tag{13}$$

The refined aligned feature $\tilde{\mathbf{F}}_k^0$ is passed to the reconstruction module for high-quality image regression (see Fig. 3).

4. Experiments

4.1. Implementation and Training Details

Configuration. As shown in Fig. 3, our network consists of four modules: feature extraction, alignment, re-weighting, and reconstruction. The feature extraction module in Sec. 3.2 contains 5 residual blocks for *all* tasks. Table 1 shows other detailed configurations, where M is the number of feature channels in the network and B is the number of residual blocks in the reconstruction module.

Training. We show the training settings in Table 1. We use 2-6 NVIDIA GeForce RTX 2080 Ti GPUs to train our models for 900K iterations for all three video restoration tasks. We adopt random vertical or horizontal flipping or 90° rotation for data augmentation. The initial learning rate is set to 5×10^{-4} and a cosine decay strategy is employed. We use Charbonnier loss for all the three tasks.

4.2. Datasets and Metrics

Video Super-Resolution. REDS [21] and Vimeo-90K [33] are two widely used datasets in Video SR. Vimeo-90K contains 64,612 training and 7,840 testing 7-frame sequences with resolution 448×256. The testing set is denoted as Vimeo-90K-T. In REDS, there are 266 training and 4 testing video sequences. Each sequence consists of 100 consecutive frames with resolution 1280×720. Following [29], we denote the testing set as REDS4. Apart from these two testing datasets, we also give the quantitative results on Vid4 [17], which consists of 4 video clips. We adopt MATLAB bicubic downsampling to generate the LR frames.

Video Deblurring. We utilize the video deblurring dataset [25] (short for VDB) to train and evaluate our models. There are a total of 61 training and 10 testing video pairs. Each pair contains blurry and sharp videos. The testing subset is marked as VDB-T. To quantitatively compare with SOTA video deblurring methods [12, 22, 29, 35], we measure the PSNR/SSIM values on the RGB channels.

Task	Video SR	Video Deblurring	Video Desnoising
Dataset	Vimeo-90K-T	VDB-T	DAVIS ($\sigma = 20$)
Baseline	37.36	29.88	35.62
+IAM	37.72 (+0.36)	32.19 (+2.31)	36.36 (+0.74)
+IAM+ARW	37.84 (+0.48)	32.28 (+2.40)	36.73 (+1.11)

Table 2. Quantitative comparison for ablation study. PSNR (dB) is reported. "Baseline" means the model without the proposed strategies. "IAM" and "ARW" denote the iterative alignment module and adaptive re-weighting, respectively.

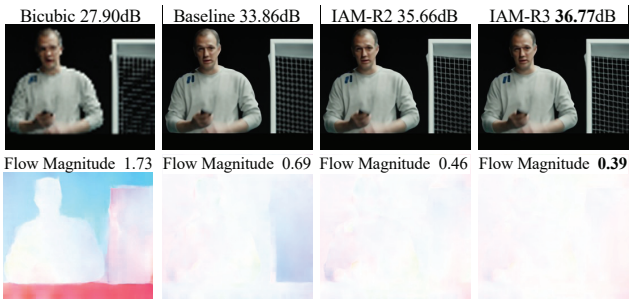


Figure 8. Analysis of the iterative number in IAM in video SR. The first line shows the predictions and the second line shows the optical flows (using RAFT [27]) between predictions and GT.

Video Denoising. In this task, we aim to remove Gaussian white noises with known noise levels (σ). Our model is trained on DAVIS [11], which contains 87 training and 30 testing 540p videos. Set8 [26] is also adopted for testing. Following [26], we keep a maximum of 85 frames for all training and testing sequences. A single denoising model is trained for all noise levels. We report our PSNR/SSIM results on the RGB channels for a fair comparison.

4.3. Analysis

In this section, we perform a comprehensive analysis of our method. We abbreviate the iterative alignment module as IAM and the adaptive re-weighting as ARW for clarity.

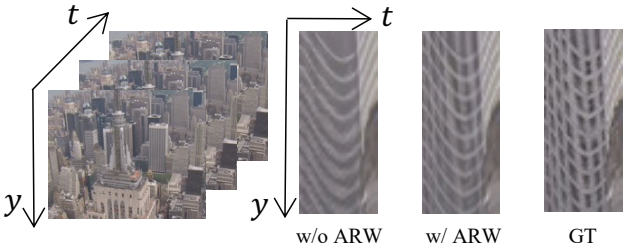
IAM and ARW. To evaluate the performance of the proposed IAM and ARW designs, we perform a quantitative comparison in Table 2. Starting from a baseline without these designs, we incrementally add the iterative alignment module (IAM) and adaptive re-weighting (ARW). As illustrated in Table 2, the proposed IAM brings about 0.36dB, 2.31dB and 0.74dB improvement on PSNR in the video SR, deblurring and denoising tasks, respectively. Besides, we notice that the utilization of ARW further pushes the PSNR up to a new height. Especially, it brings more improvement in the denoising task. All these results manifest the effectiveness of our proposed IAM and ARW strategies.

Iterative Number in IAM. We assess the influence of the iterative number in Table 3 on video SR. Compared to the baseline that performs a single prediction of each sub-alignment (identical to the progressive alignment), we gradually increase the number of refinements to 2 and 3 (de-

Methods		PSNR (dB)	SSIM	Runtime (ms)
Baseline		37.36	0.9468	153
IAM	R2	37.68 (+0.32)	0.9487	166
	R3	37.72 (+0.36)	0.9490	169
ARW	Acc.	37.39 (+0.03)	0.9469	154
	Con.	37.43 (+0.07)	0.9469	158
Full		37.84 (+0.48)	0.9498	170

Table 3. Ablation study on different IAM and ARW settings for video SR. The running time of each model is also reported with an input size of $7 \times 3 \times 64 \times 112$.

(a) Visual comparison on video SR.



(b) Visual analysis for tempoal profile on video SR.

Figure 9. Analysis of the ARW module in Video SR. (a) Visual comparison between without (w/o) and with (w/) ARW. (b) Temporal consistency comparison between without and with ARW.

noted as R2 and R3) for sub-alignments, resulting in PSNR gains by 0.32dB and 0.36dB, respectively. It is noteworthy that the increase of running time is quite minor (13-16ms). Also, as illustrated in Fig. 8, the optical flow between the prediction and GT becomes smaller with the increase of refinements, indicating more accurate alignment. Both quantitative and qualitative results suggest that our IAM can significantly improve the alignment accuracy by reducing the error accumulation during propagation.

Re-weighting Type in ARW. As shown in Table 3, we study the proposed accuracy- and consistency-based re-weighting strategies for video SR. Compared with the baseline, the accuracy-based re-weighting leads to a 0.03dB gain while the consistency one obtains 0.07dB improvement, only costing extra **1-5ms**. Figure 9 shows some examples to illustrate the improved accuracy and consistency of our ARW. It can be observed that the model with our re-weighting module is able to restore more accurate textures while maintaining temporal consistency.

4.4. Comparison with State-of-The-Art Methods

We compare our method with state-of-the-art approaches quantitatively and qualitatively in the video SR, video deblurring and video denoising tasks.

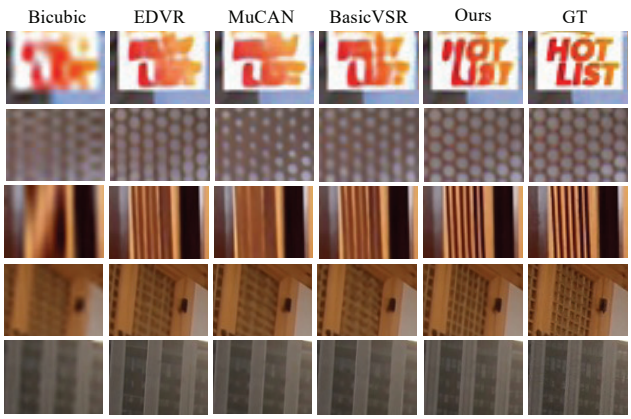


Figure 10. Qualitative comparison on Vimeo-90K-T [33] and REDS4 [21] in video SR.

Method	REDS4 (RGB)		Vid4 (Y)	
	N	PSNR/SSIM	N	PSNR/SSIM
Bicubic	1	26.14/0.7292	1	23.78/0.6347
TOF [33]	5	27.98/0.7990	7	25.89/0.7651
DUF [10]	5	28.63/0.8251	7	27.33/**0.8318**
EDVR [29]	5	31.09/0.8800	7	27.35/0.8264
MuCAN [14]	5	30.88/0.8750	7	27.26/0.8215
VSR-T [2]	5	**31.19/0.8815**	7	27.36/0.8258
*IconVSR [3]	5	30.81/0.8746	7	**27.39**/0.8279
Ours	5	**31.30/0.8850**	7	**27.90/0.8380**

Table 4. REDS4 [21] and Vid4 [17] results under the $\times 4$ setting in video SR. The PSNR(dB)/SSIM results are evaluated under x4 setting. '$*$' indicates the results are from [2], that trains and evaluates IconVSR under 5/7-frame settings.

Video Super-resolution. Table 5 and Table 4 exhibit the quantitative results of our method and existing video SR methods [2–4, 7, 10, 14, 29, 33] on Vimeo-90K-T [33], REDS4 [21] and Vid4 [17] datasets. Compared to the representative independent [29] and progressive (BasicVSR [3]) alignment methods, our method obtains superior Y-channel PSNR performance with 0.23dB and 0.66dB improvement on Vimeo-90K-T, respectively. In addition, our model also surpasses the VSR-T [2] by 0.13dB, which has much more parameters. The results of BasicVSR++ on Vimeo-90K-T are obtained by pre-trianing on REDS. Though our model is only trained on Vimeo-90K without pre-training (as a typical setup), our model still performs better than it. In terms of the Vid4 [17] dataset, our method achieves significant improvement with **0.51dB** on PSNR compared with IconVSR [3]. Note that we only include recent methods which employ 5/7-frame settings for a fair comparison on REDS4 and Vid4. Fig. 10 shows the visual comparison on Vimeo-90K-T and REDS4. Our model recovers much clearer text and more accurate structures compared to other methods.

Video Denoising. Following previous methods [1, 5, 26], we adopt Set8 [26] and DAVIS [11] as our benchmarks in

Methods	Bicubic	EDVR [29]	MuCAN [14]	BasicVSR [3]	IconVSR [3]	†BasicVSR++ [4]	VSR-T [2]	Ours
nFrame	1	1	7	7	7	7	7	7
Param.	-	20.6M	19.8M	6.3M	8.7M	7.3M	43.8M	17.0M
RGB	29.79/0.8483	35.79/0.9374	-	-	-	-	**35.88/0.9380**	**35.96/0.9389**
Y	31.32/0.8684	37.61/0.9489	37.32/0.9465	37.18/0.9450	37.47/0.9476	**37.79/0.9500**	37.71/0.9494	**37.84/0.9498**

Table 5. Vimeo-90K-T [33] results in video SR. The PSNR(dB)/SSIM results are obtained under the $\times 4$ setting. Numbers in **red** and **blue** refer to the best and second-best results. '†' means BasicVSR++ uses an **additional REDS dataset** for pre-training.

Dataset	σ	VNLB [1]	V-BM4D [20]	VNLnet [5]	FastDVD [26]	Ours
Set8	10	**37.26**	36.05	37.10	36.44	**37.25**
	20	33.72	32.19	**33.88**	33.43	**34.05**
	30	**31.74**	30.00	-	31.68	**32.19**
	40	30.39	28.48	**30.55**	30.46	**30.89**
	50	29.24	27.33	29.47	**29.53**	**29.90**
DAVIS	10	**38.85**	37.58	35.83	38.71	**39.75**
	20	35.68	33.88	34.49	**35.77**	**36.73**
	30	33.73	31.65	-	**34.04**	**34.89**
	40	32.32	30.05	32.32	**32.82**	**33.56**
	50	31.13	28.80	31.43	**31.86**	**32.51**

Table 6. Set8 [26] and DAVIS [11] results in video denoising. PSNR(dB) results are reported.

Meth.	EDVR [29]	STFA [35]	Pan [22]	ARVo [12]	Ours-M	Ours
Param.	23.6M	5.4M	16.2M	-	12.7M	16.7M
PSNR	28.51	31.24	32.13	**32.80**	32.28	**32.92**
SSIM	0.864	0.934	0.927	0.935	**0.942**	**0.948**

Table 7. VDB-T [25] results in video deblurring. "Ours-M" and "Ours" denote our medium and standard models.

the video denoising task. The quantitative results are reported in Table 6. Our model achieves the best results under most noise levels. Especially, compared with the second-best approaches, our method largely improves the PSNR by **0.37dB** and **0.65dB** under the noise level $\sigma = 50$ on Set8 and DAVIS, respectively. Figure 11 presents some qualitative results. It is observed that our method restores richer and clearer textures compared with other approaches.

Video Deblurring. We compare our method with several recent video deblurring approaches [8, 22, 25, 29, 35] on VDB-T [25]. As illustrated in Table 1, two models with different sizes (10 or 40 residual blocks) are developed, denoted as "Ours-M" and "Ours". From Table 7, compared to the second best ARVo [12], we see that our model achieves **0.12dB** and 0.013 improvement on PSNR and SSIM, respectively. Some visual examples illustrated in Figure 12 also demonstrate that our model is able to handle some challenging cases with complex motion blur.

Limitation & Societal Impact The proposed design are mainly for improving the accuracy of the long-range alignment. There remains plenty of room to optimize the modeling of subtle motion. Besides, further improving the efficiency of the entire pipeline is also our future target. All of our models are trained and evaluated using public available video restoration datasets, presenting no potentially negative societal impacts.

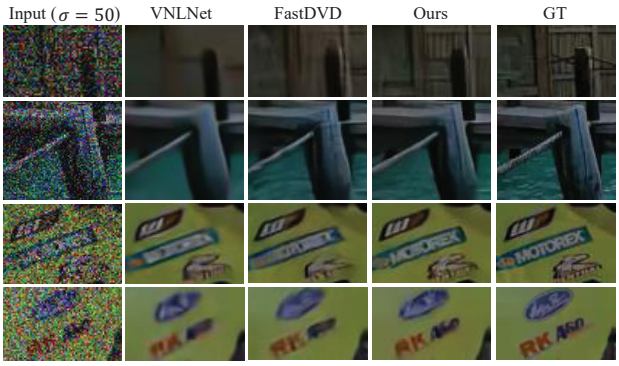


Figure 11. Qualitative comparison on Set8 [26] in video denoising.

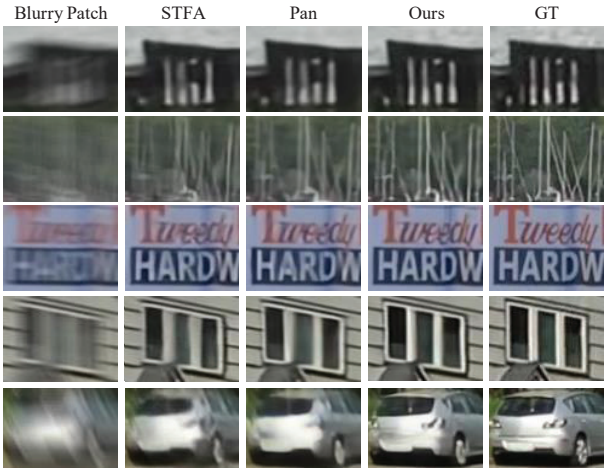


Figure 12. Visual results on VDB-T [25] in video deblurring.

5. Conclusion

In this paper, we propose a simple yet effective iterative alignment algorithm (IAM) and an efficient adaptive reweighting strategy (ARW) to better utilize multi-frame information. The quantitative and qualitative results of three video restoration tasks illustrate the effectiveness of our method. Besides, we show that our method is general and can be deployed in existing video processing systems to further improve their performance. We will explore more video-based tasks in the future. The code will be publicly available to promote the development of the community.

Acknowledgment GPUs supported by SmartMore Corporation and the Information Technology Services Office (ITSO) at the Chinese University of Hong Kong, Shenzhen.

References

[1] Pablo Arias and Jean-Michel Morel. Video denoising via empirical bayesian estimation of space-time patches. *Journal of Mathematical Imaging and Vision*, 60(1):70–93, 2018. 7, 8

[2] Jiezhang Cao, Yawei Li, Kai Zhang, and Luc Van Gool. Video super-resolution transformer. *arXiv*, 2021. 4, 5, 7, 8

[3] Kelvin C.K. 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*, 2021. 1, 2, 3, 4, 5, 7, 8

[4] Kelvin CK Chan, Shangchen Zhou, Xiangyu Xu, and Chen Change Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022. 3, 4, 7, 8

[5] Xinyuan Chen, Li Song, and Xiaokang Yang. Deep rnns for video denoising. In *Applications of Digital Image Processing XXXIX*, volume 9971, page 99711T. International Society for Optics and Photonics, 2016. 7, 8

[6] 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*, pages 764–773, 2017. 5

[7] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3897–3906, 2019. 7

[8] Tae Hyun Kim, Kyoung Mu Lee, Bernhard Scholkopf, and Michael Hirsch. Online video deblurring via dynamic temporal blending network. In *CVPR*, pages 4038–4047, 2017. 8

[9] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In *European Conference on Computer Vision*, pages 645–660. Springer, 2020. 2, 3, 5

[10] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *CVPR*, pages 3224–3232, 2018. 7

[11] Anna Khoreva, Anna Rohrbach, and Bernt Schiele. Video object segmentation with language referring expressions. In *Asian Conference on Computer Vision*, pages 123–141. Springer, 2018. 6, 7, 8

[12] Dongxu Li, Chenchen Xu, Kaihao Zhang, Xin Yu, Yiran Zhong, Wenqi Ren, Hanna Suominen, and Hongdong Li. Arvo: Learning all-range volumetric correspondence for video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7721–7731, 2021. 6, 8

[13] Feng Li, Huihui Bai, and Yao Zhao. Learning a deep dual attention network for video super-resolution. *IEEE Transactions on Image Processing*, 29:4474–4488, 2020. 2, 3, 5

[14] Wenbo Li, Xin Tao, Taian Guo, Lu Qi, Jiangbo Lu, and Jiaya Jia. Mucan: Multi-correspondence aggregation network for video super-resolution. In *ECCV*, pages 335–351. Springer, 2020. 3, 4, 7, 8

[15] Wenbo Li, Kun Zhou, Lu Qi, Nianjuan Jiang, Jiangbo Lu, and Jiaya Jia. Lapar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *NeurIPS*, 33, 2020. 2

[16] Wenbo Li, Kun Zhou, Lu Qi, Liying Lu, and Jiangbo Lu. Best-buddy gans for highly detailed image super-resolution. *AAAI Conference on Artificial Intelligence*, 2022. 2

[17] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. *IEEE transactions on pattern analysis and machine intelligence*, 36(2):346–360, 2013. 6, 7

[18] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2507–2515, 2017. 3

[19] Zhi-Song Liu, Li-Wen Wang, Chu-Tak Li, Wan-Chi Siu, and Yui-Lam Chan. Image super-resolution via attention based back projection networks. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3517–3525. IEEE, 2019. 2, 3, 5

[20] Matteo Maggioni, Giacomo Boracchi, Alessandro Foi, and Karen Egiazarian. Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. *IEEE Transactions on image processing*, 21(9):3952–3966, 2012. 8

[21] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 4, 6, 7

[22] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3043–3051, 2020. 2, 6, 8

[23] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4161–4170, 2017. 3

[24] Yaniv Romano, John Isidoro, and Peyman Milanfar. Raisr: rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, 2016. 3

[25] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. 6, 8

[26] Matias Tassano, Julie Delon, and Thomas Veit. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In *CVPR*, pages 1354–1363, 2020. 3, 6, 7, 8

[27] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 4, 6

[28] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video

super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369, 2020. 2, 3

[29] Xintao Wang, Kelvin CK 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*, pages 0–0, 2019. 1, 2, 3, 4, 5, 6, 7, 8

[30] Bartlomiej Wronski, Ignacio Garcia-Dorado, Manfred Ernst, Damien Kelly, Michael Krainin, Chia-Kai Liang, Marc Levoy, and Peyman Milanfar. Handheld multi-frame super-resolution. *ACM Transactions on Graphics (TOG)*, 38(4):1–18, 2019. 2, 3

[31] Junru Wu, Xiang Yu, Ding Liu, Manmohan Chandraker, and Zhangyang Wang. David: Dual-attentional video deblurring. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2376–2385, 2020. 2, 3, 5

[32] Xiangyu Xu, Muchen Li, and Wenxiu Sun. Learning deformable kernels for image and video denoising. *arXiv preprint arXiv:1904.06903*, 2019. 3

[33] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. *IJCV*, 127(8):1106–1125, 2019. 1, 2, 3, 6, 7, 8

[34] Ruofan Zhou and Sabine Susstrunk. Kernel modeling super-resolution on real low-resolution images. In *ICCV*, pages 2433–2443, 2019. 3

[35] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, pages 2482–2491, 2019. 3, 6, 8