Abstract

We consider the problem of space-time super-resolution (ST-SR): increasing spatial resolution of video frames and simultaneously interpolating frames to increase the frame rate. Modern approaches handle these axes one at a time. In contrast, our proposed model called STARnet super-resolves jointly in space and time. This allows us to leverage mutually informative relationships between time and space: higher resolution can provide more detailed information about motion, and higher frame-rate can provide better pixel alignment. The components of our model that generate latent low- and high-resolution representations during ST-SR can be used to finetune a specialized mechanism for just spatial or just temporal SR. Experimental results demonstrate that STARnet improves the performances of space-time, spatial, and temporal video SR by substantial margins on publicly available datasets.

1. Introduction

The goal of Space-Time Super-Resolution (ST-SR), originally proposed by [49], is to transform a low spatial resolution video with a low frame-rate to a video with higher spatial and temporal resolutions. However, existing SR methods treat spatial and temporal upsampling independently. Space SR (S-SR) with multiple input frames, (i.e., multi-image SR [11, 12] and video SR [22, 43, 46, 17]), aims to super-resolve spatial low-resolution (S-LR) frames to spatial high-resolution (S-HR) frames by spatially aligning similar frames (Fig. 1 (a)). Time SR (T-SR) aims to increase the frame-rate of input frames from temporal low-resolution (T-LR) frames to temporal high-resolution (T-HR) frames by temporally interpolating in-between frames [45, 56, 32, 41] (Fig. 1 (b)).

While few ST-SR methods are presented [49, 50, 47, 40, 32], these methods are not learning-based method and require each input video to be long enough to extract meaningful space-time patterns. [48] proposed ST-SR based on a deep network. However, this method fails to fully exploit the advantages of ST-SR schema because it relies only on LR for interpolation.

On the other hand, one can perform ST-SR by using any learning-based S-SR and T-SR alternately and independently. For example, in-between frames are constructed on S-LR, and then their SR frames are produced by S-SR; Fig. 1(c). The other way around is to spatially upsample input frames by S-SR, and then to perform T-SR to construct their in-between frames; Fig. 1(d).

However, space and time are obviously related. This relation allows us to jointly employ spatial and temporal representations for solving vision tasks on both human [20, 21, 8] and machine perceptions [39, 62, 6, 55, 17, 57, 31]. Intuitively, more accurate motions can be represented on a higher spatial representation and, the other way around, a higher temporal representation (i.e., more frames all of which are similar in appearance) can be used to accurately extract more spatial contexts captured in the temporal frames as done in multi-image SR and video SR. This intuition is also supported by various joint learning problems [13, 15, 61, 1, 60, 56, 29], which are proven to improve learning efficiency and prediction accuracy.

In order to utilize the complementary nature of space and time, we propose the Space-Time-Aware multiResolution Network, called STARnet. STARnet explicitly incorporates spatial and temporal representations for augmenting S-SR and T-SR mutually in LR and HR spaces by presenting direct connections from LR to HR for ST-SR, indicated as purple arrows in Fig. 1(e). This network also provides the extensibility where the same network can be further finetuned for either of ST-SR, S-SR, or T-SR. As shown in Fig. 2, STAR-based finetuned models perform better than state-of-the-arts [58, 14, 3, 17].

The main contributions of this paper are as follows:

1) The novel learning-based ST-SR method, which trains a deep network end-to-end to jointly learn spatial and temporal contexts, leading to what we call Space-Time-Aware multiResolution Networks (STARnet). This approach outperforms the combinations of S-SR and T-SR methods.

2) Joint learning on multiple resolutions to estimate both
large and subtle motions observed in videos. Performing T-SR on S-HR frames has difficulties in estimating large motions, while subtle motions can be difficult to interpolate on S-LR frames. Our joint learning solves both problems by presenting rich multi-scale features via direct lateral connections between multiple resolutions.

3) A novel view of S-SR and T-SR that are superior to direct S-SR and T-SR. In contrast to the direct S-SR and T-SR approaches, our S-SR and T-SR models are acquired by finetuning STAR. This finetuning from STAR allows the S-SR and T-SR models to be augmented by ST-SR learning; (1) S-SR is augmented by interpolated frames as well as by input frames and (2) T-SR is augmented by subtle motions observed in S-HR as well as large motion observed in S-LR.

2. Related Work

Space SR. Deep SR [9] is extended by better up-sampling layers [51], residual learning [26, 54], back-projection [14, 16], recursive layers [27], and progressive upsampling [30]. In video SR, temporal information is retained by frame concatenation [7, 24] and recurrent networks [22, 46, 17].

Time SR. T-SR, or video interpolation, aims to synthesize in-between frames [46, 45, 23, 55, 42, 41, 3, 43, 37, 59]. The previous methods use a flow image as a motion representation [23, 41, 3, 58, 59]. However, the flow image suffers from blur and large motions. DAIN [3] employed monocular depth estimation in order to support robust flow estimation. As another approach, by spatially downscaling input S-HR frames, large and subtle motions can be extracted in downscaled S-LR and input S-HR frames, respectively [37, 43]. While these methods [37, 43] downscale input S-HR frames for T-SR with joint training of multiple spatial resolutions, STARnet upscals input S-LR frames both in input and interpolated frames for ST-SR with joint training of multiple spatial and temporal resolutions.

Space-Time SR. The first work of ST-SR [49, 50] solved huge linear equations, then created a vector containing all the space-time measurement from all LR frames. Later, [47] presented ST-SR from a single video recording under the assumption of spatial and temporal recurrences. These
previous work \[49, 50, 47, 32, 40\] have several drawbacks, such as dependencies between the equations, its sensitivity to some parameters, and required longer videos to extract meaningful space-time patterns. \[43\] proposed STSR method to learn LR-HR non-linear mapping. However, it did not investigate the effectiveness of multiple spatial resolutions to improve the ST-SR results. Furthermore, it is also evaluated on a limited test set.

Another approach is to combine S-SR and T-SR, as shown in Fig. 1(c) and (d). However, this approach treats each context, spatial and temporal, independently. ST-SR has not been investigated thoroughly using joint learning.

### 3. Space-Time-Aware multiResolution

#### 3.1. Formulation

Given two LR frames ($I_t^L$ and $I_{t+1}^L$) with size of ($M^L \times N^L$), ST-SR obtains space-time SR frames ($I_t^H$, $I_{t+1}^H$, $I_t^{H+}$) with size of ($M^H \times N^H$) where $n \in [0, 1]$ and $M^L < M^H$ and $N^L < N^H$. The goal of ST-SR is to produce ($I_t^H$, $I_{t+1}^H$, $I_t^{H+}$) from ($I_t^L$, $I_{t+1}^L$), where $T+$ indicates the higher number of frames than $T$. In addition, STARNet computes an inequation between S-LR frame ($I_{t+1}^L$) from ($I_t^L$, $I_{t+1}^L$) for joint learning on LR and HR in space and time. Bidirectional dense motion flow maps, $F_{t\rightarrow t+1}$ and $F_{t+1\rightarrow t}$ (describing a 2D vector per pixel), between $I_t^L$ and $I_{t+1}^L$ are precomputed. $L_t \in \mathbb{R}^{M^L \times N^L \times c^L}$ and $H_t \in \mathbb{R}^{M^H \times N^H \times c^H}$ represent the S-LR and S-HR feature-maps on time $t$, respectively, where $c^L$ and $c^H$ are the number of channels.

STARNet’s operation is divided into three stages: initialization (stage 1), refinement (stage 2), and reconstruction (stage 3); Fig. 1. We train the entire network end-to-end.

**Initialization (Stage 1)** achieves joint learning of S-SR, T-SR, and ST-SR on LR and HR where T-SR and ST-SR are performed in the same subnetwork indicated by “ST-SR.” This stage takes four inputs: two RGB frames ($I_t^L$, $I_{t+1}^L$) and their bidirectional flow images ($F_{t\rightarrow t+1}$, $F_{t+1\rightarrow t}$). Stage 1 is defined as follows:

**S-SR:**

$t: H_t^L = \text{Net}_S(I_t^L, I_{t+1}^L, F_{t\rightarrow t+1}; \theta)$

$L_t^L = \text{Net}_D(H_t^L; \theta_d)$

$L_{t+1}^L = \text{Net}_D(H_{t+1}^L; \theta_d)$

**Motion:**

$H_t^L, L_{t+1}^L = \text{Net}_M(F_{t\rightarrow t+1}, F_{t+1\rightarrow t}; \theta_m)$

**ST-SR:**

$t+n: H_{t+n}^L = \text{Net}_S(H_t^L, H_{t+1}^L, L_t^L, L_{t+1}^L, M; \theta)$

$L_{t+n}^L = \text{Net}_D(H_{t+n}^L; \theta_d)$

In S-SR, S-HR feature-maps ($H_t$ and $H_{t+1}$) are produced by Net$_S$, as expressed in Eq. (1). As with other video SR methods, this S-SR is performed with sequential frames ($I_t^L$ and $I_{t+1}^L$) and their flow image ($F_{t\rightarrow t+1}$ or $F_{t+1\rightarrow t}$). $\theta$ denotes a set of weights in each network. Following up- and down-samplings for enhancing features for SR \[14, 17\], $H_t$ and $H_{t+1}$ are downscaled by Net$_D$ for updating $L_t$ and $L_{t+1}$, respectively, as expressed in Eq. (2). Net$_M$ produces a motion representation ($M$) which is calculated from the bidirectional optical flows; Eq. (3). The output of Net$_S$ is flow feature maps, learned by a CNN. While it is hard to interpret these features directly, they are intended to help spatial alignment between $F_{t\rightarrow t+1}$ and $F_{t+1\rightarrow t}$.

Finally, with the concatenation of all these features, ST-SR in the feature space is performed by Net$_{ST}$; Eq. (4). Net$_{ST}$ achieves T-SR as well as ST-SR which are incorporated on LR and HR, shown as blue and purple arrows in Fig. 1(e). The outputs of stage 1 are HR and LR feature-maps ($H_{t+n}^L$ and $L_{t+n}^L$) for an in-between frame.

In this stage, STARNet maintains cycle consistencies (1) between S-HR and S-LR and (2) between $t$ and $t+1$, while such a cycle consistency is demonstrated for general purposes \[64, 13, 63\].

**Refinement (Stage 2)** further maintains the cycle consistencies for refining the feature-maps again. While raw optical flows ($F_{t\rightarrow t+1}$ and $F_{t+1\rightarrow t}$) are used in Eq. (1) of Stage 1, the motion feature ($M$) is used in the first equations of Eqs \[5, 7, 9\], and \[10\] in Stage 2. This difference allows us to produce more reliable feature-maps. For further refinement, residual features are extracted in Eqs. \[9, 8\], and \[11\], as proposed in RBPN \[17\] for precise spatial alignment of temporal features.

Finally, Stage 2 is defined as follows:

$t: H_t^b = \text{Net}_B(L_{t+n}, L_t, M; \theta_b)$

$L_t^f = \text{Net}_D(H_t^b; \theta_d)$

$\hat{H}_t = H_t + \text{ReLU}(H_t - H_t^b)$

$\hat{L}_t = L_t + \text{ReLU}(L_t - L_t^b)$

$H_{t+1}^f = \text{Net}_F(L_{t+n}, L_{t+1}, M; \theta_f)$

$L_{t+1}^f = \text{Net}_D(H_{t+1}^f; \theta_d)$

$\hat{H}_{t+1} = H_{t+1} + \text{ReLU}(H_{t+1} - H_{t+1}^f)$

$\hat{L}_{t+1} = L_{t+1} + \text{ReLU}(L_{t+1} - L_{t+1}^f)$

$t+n: H_{t+n}^f = \text{Net}_F(L_t, L_{t+n}, M; \theta_f)$

$L_{t+n}^f = \text{Net}_D(H_{t+n}^f; \theta_d)$

$H_{t+n}^b = \text{Net}_B(L_{t+1}, L_{t+n}, M; \theta_b)$

$L_{t+n}^b = \text{Net}_D(H_{t+n}^b; \theta_d)$

$\hat{H}_{t+n} = H_{t+n} + \text{ReLU}(H_{t+n} - H_{t+n}^f) + \text{ReLU}(H_{t+n} - H_{t+n}^b)$

$\hat{L}_{t+n} = L_{t+n} + \text{ReLU}(L_{t+n} - L_{t+n}^f) + \text{ReLU}(L_{t+n} - L_{t+n}^b)$

**Reconstruction (Stage 3)** transforms four feature-maps ($H_t$, $\hat{H}_{t+n}$, $\hat{H}_{t+1}$, and $L_{t+n}$) to their corresponding images ($I_{t+1}^H$, $I_{t+1}^H$, $I_{t+n}^H$, and $I_{t+n}^L$) by using only one conv layer Net$_{rec}$; for example, $I_t^H = \text{Net}_{rec}(\hat{H}_t; \theta_{rec})$. 

---

\[\text{Eq. (1)}\]

\[\text{Eq. (2)}\]

\[\text{Eq. (3)}\]

\[\text{Eq. (4)}\]

\[\text{Eq. (5)}\]

\[\text{Eq. (6)}\]

\[\text{Eq. (7)}\]

\[\text{Eq. (8)}\]

\[\text{Eq. (9)}\]

\[\text{Eq. (10)}\]

\[\text{Eq. (11)}\]
First, S-SR produces a pair of S-LR and S-HR feature-maps \((L_t, H_t, L_{t+1}, \text{and} \ H_{t+1})\) at each time. Motion representation \((M)\) is calculated by Motion network from bidirectional optical flow images \((F_{t\rightarrow t+1} \text{ and } F_{t+1\rightarrow t})\). With these features, ST-SR produces the feature-maps of the in-between frame \((L_{t+n} \text{ and } H_{t+n})\). Finally, we reconstruct all outputs of STARnet \((I_{sr t}, I_{sr t+n}, I_{sr t+1}, \text{and} \ I_{lt+n})\) by concatenating all features-maps on LR and HR in space and time.

3.2. Training Objectives

The reconstructed images of STARnet \((I_{sr t}, I_{sr t+n}, I_{sr t+1}, \text{and} \ I_{lt+n})\) are compared with their ground-truth images by loss functions in a training phase. For this training, (1) S-HR images as the ground-truth images are downscaled to S-LR images and (2) T-HR frames as the ground-truth frames are skimmed to T-LR frames. The loss functions are divided into the following three types:

- **Space loss** is evaluated on \(I_{sr t}\) and \(I_{sr t+1}\).
- **Time loss** is evaluated only on \(I_{lt+n}\).
- **Space-Time loss** is evaluated only on \(I_{lt+n}\).

Our framework provides the following four variants, which are trained with different training objectives.

- **STAR** is trained using all of the aforementioned three losses on LR and HR in space and time. STAR produces \{\(I_{sr t}\)\} from \(t=0\) to \(T\) and \{\(I_{lt+n}\)\} simultaneously as in Fig. 4(a).
- **STAR-ST** is a fine-tuned model from STAR using Space and Space-Time losses on HR in space and time. The network is optimized on the space-time super-resolved frames \{\(I_{sr t}\)\} from \(t=0\) to \(T\) as in Fig. 4(b).
- **STAR-S** is a fine-tuned model from STAR using Space loss on S-HR, optimizing only \{\(I_{sr t}\)\} from \(t=0\) to \(T\) as in Fig. 4(c).
- **STAR-T** is a fine-tuned model from STAR using Time loss on T-HR as in Fig. 4(d). STAR-T can be trained on two different regimes, S-LR and S-HR. While STAR-T uses the original frames (S-HR) as input frames, STAR-T uses the downscaled frames (S-LR) as input frames.

3.3. Loss Functions

Each of Space, Time, and Space-Time losses consists of two types of loss functions. \(L_1\) and \(L_{vgg}\). \(L_1\) is the loss per-pixel between a predicted super-resolved frame \((I_{sr t})\) and its ground-truth HR frame \((I_{ht})\) where \(t \in [T]\).

\[
L_1 = \sum_{t=0}^{T} ||I_{ht} - I_{sr t}||_1 \tag{12}
\]

\(L_{vgg}\) is calculated in the feature space using a pretrained VGG19 network \([52]\). For computing \(L_{vgg}\), both \(I_{ht}\) and \(I_{sr t}\) are mapped into the feature space by differentiable functions \(f_m\) from the VGG multiple max-pool layer \((m = 5)\).

\[
L_{vgg} = \sum_{t=0}^{T} ||f_m(I_{ht}) - f_m(I_{sr t})||_2^2 \tag{13}
\]

\(L_1\) is for fulfilling standard image quality assessment metrics such as PSNR and validated for SR \([42, 5]\), while...
$L_{vgg}$ improves visual perception \cite{25, 10}. Based on this fact, only $L_1$ or a weighted sum of $L_1$ and $L_{vgg}$ is utilized for training STARnet depending on the purpose.

### 3.4. Flow Refinement

As mentioned in Section 3.1, we use flow images pre-computed by \cite{34}. As revealed in many video interpolation papers \cite{36, 45, 23, 35, 42, 41, 3, 43, 37, 59}, large motions between $t$ and $t+1$ make video interpolation difficult. Flow noise due to such large motions has a bad effect on the interpolation results. While STARnet suppresses this bad effect by T-SR not only in S-HR but also in S-LR, it is difficult to fully resolve this problem. For further improvement, we propose a simple solution to refine or denoise the flow images, called a Flow Refinement (FR) module.

Let $F_{t\rightarrow t+1}$ and $F_{t+1\rightarrow t}$ are flow images between frames $I^t_t$ and $I^t_{t+1}$ on forward and backward motions, respectively. During training, $F_{t\rightarrow t+n}$ can be calculated from an input frame at $t$ to the ground truth (i.e., from $I^t_t$ to $I^{t+n}_{t+n}$). Net$\_flow$ is a U-Net which defines as follows.

\[
\hat{F}_{t\rightarrow t+1} = \text{Net}_{flow}(F_{t\rightarrow t+1}, I_t, I_{t+1}; \theta_{flow})
\]

\[
\hat{F}_{t+1\rightarrow t} = \text{Net}_{flow}(F_{t+1\rightarrow t}, I_{t+1}, I_t; \theta_{flow})
\]

To reduce the noise, we propose the following flow refinement loss.

\[
L_{flow} = \|\hat{F}_{t\rightarrow t+1}-(F_{t\rightarrow t+1}+F_{t+1\rightarrow t})\|_2^2
\]

\[
+\|\hat{F}_{t+1\rightarrow t}-(F_{t\rightarrow t+1}+F_{t+1\rightarrow t})\|_2^2
\]

With $L_{flow}$, the loss functions for training STARnet are defined as follows:

\[
L_r = w_1 \ast L_1 + w_2 \ast L_{flow}
\]

\[
L_f = L_r + w_3 \ast L_{vgg}
\]

### 4. Experimental Results

In all experiments, we focus on 4× SR factor and $n = 0.5$. $I_{t^r}$ and $I_{t^w}$ denote the SR frames of input frames and in-between frames, respectively.

#### 4.1. Implementation Details

**Stage 1.** For Net$_S$ and Net$_P$, we use DBPN \cite{14} or RBPN \cite{12} that have up- and down-sampling layers to simultaneously produce a pair of S-LR and S-HR features with $c^0=64$ and $c^2=128$. Net$_M$ is constructed with two residual blocks where each block consists of two conv layers with $3 \times 3$ with stride = 1 and pad by 1. Net$_{ST}$ has five residual blocks followed by deconv layers for upsampling.

**Stage 2.** Both Net$_P$ and Net$_B$ are constructed using five residual blocks and deconv layers.

**Train Dataset.** We use the triplet training set in Vimeo90K \cite{58} for training. This dataset has 51,313 triplets from 14,777 video clips with a fixed resolution, 448 × 256. During training, we apply augmentation, such as rotation, flipping, and random cropping. The original images are regarded as S-HR and downsampled to 112 × 64 S-LR frames (4× smaller than the originals) with Bicubic interpolation.

#### 4.2. Ablation Studies

Here, we evaluate STARnet without T-SR paths (blue arrows in Fig. 1(e)) in order to clarify the effectiveness our core contribution (i.e., joint learning in time and space on multiple resolutions) with a simplified network using direct ST-SR paths (purple arrows). The test set of Vimeo90K \cite{58} is used.

**Basic components.** We evaluate the basic components on STARnet. In the first experiment, we remove the refinement part (i.e., Stage 2), leaving only the initialization part. Second, we omit input flow images and Net$_{flow}$, so no motion context is used (STAR w/o Flow). Third, the FR module is removed. Finally, the full model is evaluated. The results of these four models are shown in “STAR w/o Stage 2,” “STAR w/o Flow,” “STAR w/o FR,” and “STAR” in Table 1. Compared with the full model, the PSNR of STAR w/o Stage 2 decreases to 0.36dB and 1.0dB on $I_{t^r}$ and $I_{t^w}$, respectively. The flow information can also improve the PSNR 0.28dB and 0.43dB on $I_{t^r}$ and $I_{t^w}$, respectively.

While FR is also useful, the quantitative improvement by FR is not substantial compared with those of the other two components. The examples of $I_{t^r}$ are shown in Fig. 3 where flow images are computed only by $I_t$ and $I_{t+1}$, only by $I_t$ and $I_{t+1}$ refined by FR, and by $I_t$ (i.e., GT in-between frame) in addition to $I_t$ and $I_{t+1}$ in (a), (b), and (c), respectively. In Fig. 5 the visual improvement by FR is substantial. This result reveals that (1) erroneous flows are
Table 1. Baseline comparison of STAR with DBPN [16] and $L_f$. Red in all tables indicates the best performance.

Table 2. Analysis on different training objectives using STARnet with DBPN [16] and $L_f$.

Table 3. Analysis on two loss functions using STARnet with RBPN [17] and $L_f$.

Table 4. Analysis on the S-SR module using STAR-ST and $L_f$.

Table 5. Analysis on larger scale T-SR (4×) on the Vimeo90K setuplet test set with $L_f$.

Table 6. Analysis on ST-SR jointly trained with T-SR with RBPN [17] and $L_f$. Models are optimized for STAR-ST w/ FR.

Larger scale T-SR. The performance on a larger scale T-SR is investigated. While the S-SR factor is the same with that in other experiments (i.e., 4×), the frame-rate is upscaled to 4×. We compare two upscaling paths: (1) STAR-ST (2× S-SR and 2× T-SR) → STAR-ST (2× S-SR and 2× T-SR) (2) STAR-ST (4× S-SR and 2× T-SR) → STAR-T (2× T-SR). For training 4× T-SR, the training set of the Vimeo90K setuplet, where each sequence has 7 frames, is used. Then, the 1st and 5th frames in each sequence are used as input frames for evaluation. As shown in Table 5, the second path is better. This result may suggest that a higher spatial resolution provides better results on T-SR.

T-SR paths on S-HR and S-LR domains. We analyze the effectiveness of T-SR on multiple spatial resolutions (blue arrows in Fig. 1(e)) as well as ST-SR (purple arrows in Fig. 1(e)). Table 6 shows the results of the following four experiments. In (1), we remove all T-SR modules (blue arrows). In (2), T-SR on S-HR is incorporated with ST-SR module. In (3), T-SR on S-LR is incorporated with ST-SR module. In (4), all modules are used as shown in Fig. 1(e). In these implementations, T-SR modules can be removed by modifying NetST in Eq. 4 so that it contains only ST-SR, ST-SR+T-SR$_{S-HR}$, ST-SR+$T-SR_{S-LR}$, and all of them for (1), (2), (3), and (4), respectively. It confirms that joint training of ST-SR and T-SR improves the performance. Both S-HR and S-LR resolutions improve the performance compared with only ST-SR, while the best results are obtained by the full STAR model.

4.3. Comparisons with State-of-the-art

The following results are obtained by the full STAR model, which is evaluated as the best in Table 6.
Table 7. Comparison on ST-SR ($I_{sr}^L$) using $L_r$. $\alpha \rightarrow \beta$ indicates the output of $\alpha$ is the input of $\beta$. Red indicates the best and blue indicates the second best performance in all tables in Section 4.3. * indicates a joint learning of RBPN and DAIN methods to perform ST-SR.

<table>
<thead>
<tr>
<th>Method</th>
<th>UCF101 PSNR</th>
<th>SSIM</th>
<th>NIQE</th>
<th>Vimeo90K PSNR</th>
<th>SSIM</th>
<th>NIQE</th>
<th>Middlebury (Other) PSNR</th>
<th>SSIM</th>
<th>NIQE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ToFlow [58] $\rightarrow$ DBPN [16]</td>
<td>27.228</td>
<td>0.885</td>
<td>9.123</td>
<td>28.821</td>
<td>0.897</td>
<td>7.758</td>
<td>24.984</td>
<td>0.790</td>
<td>6.473</td>
</tr>
<tr>
<td>DBPN [16] $\rightarrow$ ToFlow [58]</td>
<td>28.112</td>
<td>0.902</td>
<td>8.630</td>
<td>29.867</td>
<td>0.915</td>
<td>7.120</td>
<td>26.012</td>
<td>0.808</td>
<td>5.801</td>
</tr>
<tr>
<td>DBPN [16] $\rightarrow$ DAIN [3]</td>
<td>28.175</td>
<td>0.902</td>
<td>8.755</td>
<td>30.021</td>
<td>0.918</td>
<td>7.223</td>
<td>26.268</td>
<td>0.809</td>
<td>5.869</td>
</tr>
<tr>
<td>DBPN-MI $\rightarrow$ DAIN [3]</td>
<td>28.578</td>
<td>0.916</td>
<td>8.922</td>
<td>30.286</td>
<td>0.923</td>
<td>7.218</td>
<td>26.447</td>
<td>0.815</td>
<td>5.702</td>
</tr>
<tr>
<td>DAIN [3] $\rightarrow$ RBPN [17]</td>
<td>27.631</td>
<td>0.909</td>
<td>8.932</td>
<td>29.422</td>
<td>0.916</td>
<td>7.253</td>
<td>25.744</td>
<td>0.811</td>
<td>5.814</td>
</tr>
<tr>
<td>RBPN [17] $\rightarrow$ DAIN [3]</td>
<td>28.729</td>
<td>0.919</td>
<td>8.769</td>
<td>30.455</td>
<td>0.926</td>
<td>7.081</td>
<td>26.766</td>
<td>0.821</td>
<td>5.522</td>
</tr>
<tr>
<td>STAR-Lf $\rightarrow$ DAIN [3]</td>
<td>28.829</td>
<td>0.920</td>
<td>8.787</td>
<td>30.608</td>
<td>0.926</td>
<td>7.081</td>
<td>26.881</td>
<td>0.824</td>
<td>4.579</td>
</tr>
<tr>
<td>STAR-ST-Lf $\rightarrow$ DAIN [3]</td>
<td>28.806</td>
<td>0.920</td>
<td>7.868</td>
<td>30.714</td>
<td>0.927</td>
<td>6.470</td>
<td>27.020</td>
<td>0.826</td>
<td>4.802</td>
</tr>
<tr>
<td>STAR-ST-Lr $\rightarrow$ DAIN [3]</td>
<td>29.111</td>
<td>0.924</td>
<td>8.787</td>
<td>30.830</td>
<td>0.929</td>
<td>7.154</td>
<td>27.115</td>
<td>0.827</td>
<td>5.423</td>
</tr>
</tbody>
</table>

Table 8. Comparison on S-SR ($I_{sr}^L$) using $L_r$.

<table>
<thead>
<tr>
<th>Method</th>
<th>UCF101 PSNR</th>
<th>SSIM</th>
<th>Vimeo90K PSNR</th>
<th>SSIM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bcubic</td>
<td>27.217</td>
<td>0.887</td>
<td>28.134</td>
<td>0.878</td>
</tr>
<tr>
<td>DBPN</td>
<td>29.828</td>
<td>0.913</td>
<td>31.505</td>
<td>0.927</td>
</tr>
<tr>
<td>DBPN-MI</td>
<td>30.666</td>
<td>0.934</td>
<td>31.835</td>
<td>0.933</td>
</tr>
<tr>
<td>RBPN</td>
<td>30.969</td>
<td>0.938</td>
<td>32.154</td>
<td>0.936</td>
</tr>
<tr>
<td>STAR-ST</td>
<td>31.532</td>
<td>0.942</td>
<td>32.547</td>
<td>0.940</td>
</tr>
<tr>
<td>STAR-S</td>
<td>31.604</td>
<td>0.943</td>
<td>32.702</td>
<td>0.941</td>
</tr>
</tbody>
</table>

ST-SR. As discussed in Section 2, older ST-SR methods [49, 50, 47, 32, 40] cannot be applied to videos in the Vimeo90K dataset. We can combine more modern S-SR and T-SR methods to perform ST-SR. We use DBPN [16] and RBPN [17] as S-SR. For T-SR, we choose ToFlow [58] and DAIN [3]. In Table 7 we present the results of ST-SR obtained by six combinations of these methods.

It is found that S-SR $\rightarrow$ T-SR performs better than T-SR $\rightarrow$ S-SR. The margin is up to 1dB on Vimeo90K, showing that the performance of previous T-SRs significantly drops on LR images. Even STAR is better than the combination of state-of-the-arts (RBPN [17] $\rightarrow$ DAIN [3]), while the best result is achieved by STAR-ST, which is the fine-tuned model from STAR. STAR-ST has a better performance around 0.38dB than RBPN [17] $\rightarrow$ DAIN [3] on Vimeo90K test set.

We can also present ST-SR as a joint learning of RBPN [17] and DAIN [3], indicated as (*). It shows that
Table 9. Comparison on T-SR on the original resolution. SSIM is almost saturated especially on UCF101, so PSNR is a better measure here. *Results are taken from Middlebury dashboard.

![Image]

Table 10. Comparison of T-SR on L-SR (l_H) with Vimeo90K [55].

![Image]

5. Conclusion

We proposed a novel approach to space-time super-resolution (ST-SR) using a deep network called Space-Time-Aware multiResolution Network (STARnet). The network super-resolves jointly in space and time. We show that a higher resolution presents detailed motions, while a higher frame-rate provides better pixel alignment. Furthermore, we demonstrate a special mechanism to improve the performance for just S-SR and T-SR. We conclude that the integration of spatial and temporal contexts is able to improve the performance of S-SR, T-SR, and ST-SR by substantial margin on publicly available datasets.

This work was supported by JSPS KAKENHI Grant Number 19K12129.
References


