Multimedia Tools and Applications
https://doi.org/10.1007/s11042-024-20000-9

SSRT: Intra- and cross-view attention for stereo image super-resolution

Qixue Yang[1] · Yi Zhang[1] · Damon M. Chandler[2] · Mylene C. Q. Farias[3]

Received: 8 June 2023 / Revised: 23 July 2024 / Accepted: 30 July 2024

Abstract

Stereo image super-resolution (SR) aims to increase the spatial resolutions of the left and right views of a stereo image in an attempt to generate higher-resolutions views that appear visually equivalent to those obtained with a higher-resolution stereo camera. The field of stereo image SR has seen rapid significant progress due in large part to the application of deep-learning-based techniques and its associated recent research advancements. Yet, despite this progress, stereo images captured under real-world conditions (e.g., using consumer-level cameras in non-laboratory settings) often contain irregular disparities between the left and right views, a fact that has not been fully considered nor properly addressed in previous works. To address this issue, in this paper, we propose a stereo image super-resolution Transformer (SSRT) network which consists of two blocks, a multi-kernel Transformer block and a cross-merging block, to fully extract intra-view features and capture cross-view dependencies. The multi-kernel Transformer block is proposed to increase the number of representation subspaces for intra-view feature extraction. The cross-merging block utilizes patch-wise attention which efficiently expands the search area to tackle stereo image pairs with arbitrary pixel offsets. Experimental results demonstrate that, for 2× stereo image super-resolution, our model with a comparable number of network parameters achieves 37.57 dB on ETH3D, 35.88 dB on Middlebury, 29.56 dB on Flickr1024, 31.52 dB on KITTI 2012, and 31.15 dB on KITTI2015 in terms of PSNR, and surpasses the state-of-the-art method by a large margin of +0.80 dB on ETH3D, +0.57 dB on Middlebury, +0.36 dB on Flickr1024, +0.14 dB on KITTI 2012, and +0.06 dB on KITTI 2015. The code is available at https://github.com/yanksx233/SSRT.

✉ Yi Zhang
yi.zhang.osu@xjtu.edu.cn

Qixue Yang
qxyang@stu.xjtu.edu.cn

Damon M. Chandler
chandler@fc.ritsumei.ac.jp

Mylene C. Q. Farias
mylene@ieee.org

[1] School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China

[2] College of Information Science and Engineering, Ritsumeikan University, Osaka 567-8570, Japan

[3] Department of Computer Science, Texas State University, San Marcos, TX 78666, USA

Published online: 09 August 2024

Keywords Stereo image · Super-resolution · Neighborhood attention · Transposed attention · Patch-wise attention · Transformer

1 Introduction

Stereo image pairs have been widely used in many stereo vision areas such as augmented reality, virtual reality and stereo video retargeting [27]. However, due to limitations in both device performance and transmission bandwidth, low resolution (LR) images may be produced, which affects not only the user experience but also the performance of other downstream vision-based algorithms. In addition, directly capturing high resolution (HR) images often requires more advanced imaging devices with more expensive electronic components, which can be economically prohibitive, and also negatively impact mobility and battery life. To address this issue, the topic of stereo image super-resolution (SR) has emerged, which aims at increasing the resolution of LR stereo image pairs to generate their HR counterparts, thereby resulting in enhanced image quality with an economical hardware overhead.

Stereo image pairs captured by a binocular camera or multiple cameras often display the same object from different viewpoints. Due to the highly similar patterns and textures between the two view images, the perceptual quality of a stereo image can possibly be enhanced by using the mutual information extracted from the stereopair. Thus, in stereo image SR, other than to utilize the dependencies between pixels in a single image (intra-view) for image reconstruction, it is also important to capture and make use of the dependencies between the two views (cross-view) to further enhance the SR performance.

1.1 Motivation

The most popular and effective way to model the intra-view dependencies is to use a so-called attention mechanism [13, 32, 42]. For example, the Transformer [4, 6, 28, 50, 54] which consists of a series of attention modules has demonstrated powerful performance in single image restoration owing to its ability to flexibly capture long-range dependencies. The attention mechanism used in the existing image restoration Transformer can be roughly classified into three categories: (1) global self-attention (e.g., IPT [4]) that views an image as a series of tokens; (2) local self-attention (e.g., SwinIR [28], HAT [6], Uformer [50]) with adaptive representation capability and a receptive field with a wider extent than that used during convolution; and (3) transposed attention (e.g., Restormer [54]) with a lighter computational cost than self-attention. Although these attention mechanisms have demonstrated powerful feature-representation capabilities, they still suffer from certain limitations. For example, token-based global self-attention often fails to model the relationship between neighboring pixels in an image, and local self-attention based on window-partitioning does not take into account the inductive bias which is helpful in enabling the attention operation to be translational and rotational equivariant. Transposed attention can be considered as an adaptive 1×1 convolution, which performs point-wise convolution over all spatial locations using a consistent kernel (i.e., a single attention weight matrix with shape $\mathbb{R}^{C \times C}$, where C denotes the number of channels). Consequently, the limited spatial receptive field and single representation subspace can potentially lead to a sub-optimal solution.

Although great success has been achieved in capturing the intra-view dependencies, it is often insufficient to utilize only the intra-view dependencies for stereo image SR because the cross-view dependencies between the two views can also be important. To capture the

cross-view dependencies to further improve the SR performance, a number of stereo image SR approaches have been presented. For example, Jeon et al. [21] manually stacked the left image and a number of right images with different horizontal shifts to model the parallax prior between the two views. However, one potential limitation of the method in [21] is that only a fixed parallax is considered. Wang et al. [43] proposed a parallax attention module by constraining the query region of a non-local method [44] to the horizontal epipolar, resulting in an algorithm that can handle large horizontal disparity with low computational complexity. Dai et al. [9] tackled both SR and disparity estimation tasks simultaneously in a unified framework via knowledge interaction among different tasks to improve the performance. Chu et al. [7] introduced a simple yet effective model for both single-view feature extraction and cross-view feature fusion via NAFNet [5] and cross attention modules, respectively. Though effective, these methods usually ignore vertical pixel offsets, while in practice stereo scenes can generally have irregular epipolar constraints. Recently, Chen et al. [3] fed concatenated features extracted from the two views into convolution layers with a larger kernel size (i.e., 7×7) to simultaneously capture both horizontal and vertical pixel offsets. However, the disparity considered in [3] is strictly constrained to the kernel size of the convolution layer because an attention mechanism is not incorporated.

1.2 Innovation

To overcome the aforementioned limitations in both intra-view and cross-view feature extraction, a stereo image SR Transformer (SSRT) network is proposed in this paper (as shown in Fig. 1). In our model, we cascade a neighborhood attention Transformer block (NATB) [15] and a dual-window transposed attention Transformer block (TATB) to form what we call a multi-kernel Transformer block (MKTB), which provides a good balance between computational cost and receptive field size. The neighborhood attention in the NATB preserves the inductive bias provided by large-kernel convolution. The dual-window partition in the TATB allows the transposed attention mechanism to generate attention kernels with different

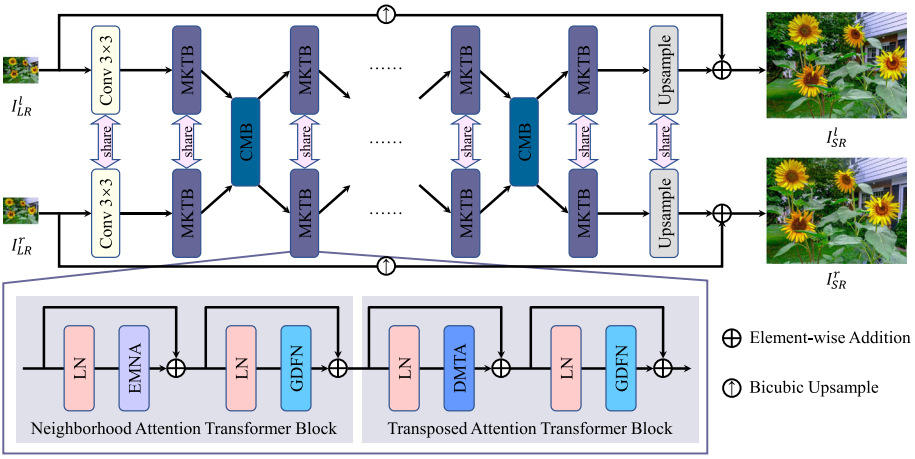

Fig. 1 Network architecture of the proposed SSRT model, which consists of multi-kernel Transformer blocks (MKTBs) and cross-merging blocks (CMBs) alternately concatenated. Note that the two branches in SSRT share the same network parameters to reduce the model complexity

weights at different spatial windows such that different image regions are represented by different feature spaces. Specifically, the TATB uses a dual-window partition mechanism within the transposed attention mechanism to enhance its feature representation capability and thus overcome the aforementioned limitation that only a single representation subspace is considered in intra-view feature extraction. Finally, to overcome the limitation that irregular disparities in real-world stereo images are not fully exploited by exiting methods, we propose a cross-merging block (CMB) to more effectively capture the irregular pixel shift between the two views in stereo images. The CMB not only fuses the left and right view features which may contain vertical pixel offsets, but it also conducts patch-matching in the LR space and patch-wise transferring in the HR space such that the large receptive field of the HR space is obtained with a relatively lower computational complexity (as shown in Fig. 2).

1.3 Contribution

Compared with existing stereo image SR methods, SSRT has several distinctive properties. First, compared with token-based global self-attention [4] and window-based local self-attention [6, 28, 50] commonly used for image super-resolution and/or restoration tasks, SSRT preserves the translational and rotational equivariance thanks to the neighborhood attention. Second, compared with conventional transposed attention adopted in [54], a dual-window partition is incorporated such that images can be represented in multiple feature spaces and thus the optimal network solution can possibly be achieved. Also, two attention blocks (i.e., NATB and TATB) are used in MKTB to balance the receptive field size and the model computational complexity. Finally, compared with most existing stereo image SR methods which employ pixel-wise attention to model cross-view dependencies, patch-wise attention is adopted in our work such that the most similar patch can be selected, thereby eliminating the consideration of irrelevant points. The more flexible image search regions

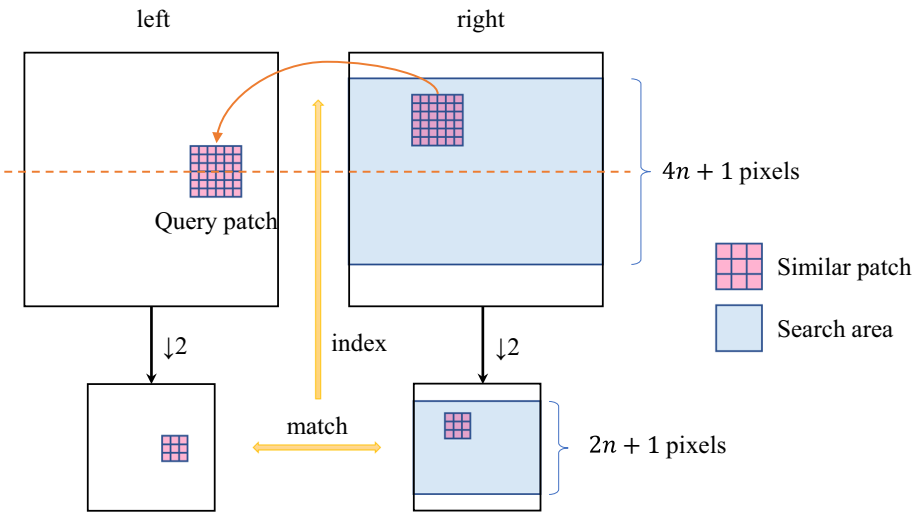

Fig. 2 An illustration of the patch-wise attention strategy used in the cross-merging block. Note that this attention only transfers the most similar patch from the search area to the query position. Also note that we conduct patch matching in the LR space and patch-wise transferring in the HR space to reduce the computational cost

and the cross-scale patch match/transfer strategy allow the model to capture both vertical and horizontal disparities between the two views with a lower computational cost. Overall, the main contributions of this paper are summarized as follows:

1. We propose to compute transposed attention in different local windows. Accordingly, we further propose a dual-window transposed attention to prevent blocking artifacts caused by using a regular window partition while still maintaining a spatially linear computational complexity.
2. We propose MKTB, which jointly utilizes adaptive large-kernel convolution and point-wise convolution for more effective modeling of intra-view dependencies.
3. We propose CMB with patch-wise attention for more effective modeling of cross-view dependencies between the two views with irregular disparities.
4. Experimental results tested on five public stereo image datasets demonstrate that the proposed model outperforms existing stereo image SR methods by a large margin with fewer/comparable network parameters.

1.4 Sections of the manuscript

The rest of the paper is organized as follows. Section 2 provides a brief review of existing image SR methods. Section 3 provides a problem formulation of the image SR task. Section 4 describes details of the proposed SSRT model. In Section 5, we analyze and discuss the performance of SSRT by using five benchmark stereo image datasets. General conclusions are presented in Section 6.

2 Related work

In this section, we provide a brief review of the existing methods for single-image and stereo image SR.

2.1 Single image super-resolution

Traditional approaches to single-image SR include the use of techniques such as neighbor embedding [2], sparse representation [53], and neighborhood regression [41]. More recently, deep-learning-based SISR methods have demonstrated impressive performance due to the strong end-to-end learning ability of the neural networks. Dong et al. [12] first introduced the use of a convolutional neural network (CNN) for SISR. Subsequently, great efforts have been made to improve CNN-based SISR by using more advanced network designs and architectures, such as residual connections [24, 25, 29], dense connections [22, 59], pixel shuffling [40], attentive networks [56], and resolution-aware networks [46]. Because PSNR-oriented methods will produce visually unrealistic images, perceptual loss [23] and adversarial loss [25, 33, 45, 57] have been exploited to further improve the details and visual qualities of the reconstructed images. Very recently, some methods achieved state-of-the-art performance by introducing various attention mechanisms, such as channel attention [10, 58], self-attention [10, 36], multi-grained attention [51], patch-matching [35], and Transformer blocks [4, 6, 8, 28].

2.2 Stereo image super-resolution

Stereo image SR aims to reconstruct the HR details from a pair of LR left and right images captured by a binocular camera. The StereoSR [21] model uses a CNN to learn a parallax prior by concatenating the left image with a number of right images that have different horizontal pixel shifts. Since StereoSR can only deal with limited parallax, Wang et al. [43] proposed a parallax-attention module to capture the dependencies between the two views by limiting the search area of a non-local method [44] to the epipolar region. Based on a parallax-attention module, Wang et al. [48] introduced a channel attention block [17] to address the intra-view problem. Zhu et al. [61] combined epipolar cross attention with an asymmetric non-local network [62] to transfer both horizontal and global contextual features from the right view to the left view. Yan et al. [52] transferred the information from the disparity domain to the image domain via a disparity alignment network. Lei et al. [26] proposed an interaction-module-based stereo image SR network composed of several interaction units with residual structures to learn the complementary information of the stereopair. Dai et al. [9] simultaneously tackled SR and disparity estimation in a unified framework to improve the performances of both tasks. Zhang et al. [60] proposed a recurrent interaction network for stereo image SR in which a recurrent interaction module was designed to learn the inter-view dependencies among the two-view multi-level features. Chen et al. [3] proposed a cross parallax attention module to address stereo image pairs with irregular epipolar lines, but the performance of the method is constrained by the size of the convolution kernel. Chu et al. [7] simplified the parallax attention module, and designed a nonlinear activation-free network for stereo image SR based on NAFNet [5]. Lin et al. [30] proposed a Transformer to efficiently capture reliable stereo correspondence and incorporate cross-view information. Liu et al. [31] proposed a coarse-to-fine cascaded parallax attention module to gradually perform parallax attention adjustments from LR to HR. A summary of existing single image and stereo image SR methods is provided in Table 1.

In the following sections, we describe our proposed SSRT model which employs multi-kernel Transformer blocks (MKTBs) and cross-merging blocks (CMBs) to simultaneously model the intra-view and cross-view dependencies for stereo image SR.

3 Problem statement

Image SR aims to reconstruct a HR image from a corresponding degraded LR image. Normally, the degradation $\mathcal{D}$ is formulated as

$$I_{LR} = \mathcal{D}(I_{HR}) = (I_{HR} * k) \downarrow_s + n, \tag{1}$$

where I_{HR} is the HR image without distortion; $*$ denotes the convolution operation; k denotes the blur kernel; $\downarrow_s$ denotes downsampling the image to $1/s$ of its original size; and n denotes noise. Generally, the degradation parameters (i.e., k, s, and n) are unknown when the images are captured. The super-resolved HR image I_{SR} can be expressed as

$$I_{SR} = \mathcal{F}(I_{LR}; \theta), \tag{2}$$

where $\mathcal{F}$ denotes an SR function to recover the LR image; and θ are the parameters associated with $\mathcal{F}$. In deep-learning-based image SR, $\mathcal{F}$ can be designed as a deep neural network, and θ denotes the corresponding network parameters. To achieve correct SR output, we train the

Table 1 A summary of existing single image and stereo image SR methods based on deep learning

Single image SR	CNN	GAN/Loss	Attention/Transformer
	Dong et al. [12]	Ledig et al. [25]	Zhang et al. [58]
	Kim et al. [24]	wang et al. [45]	Dai et al. [10]
	Lim et al. [29]	Zhang et al. [57]	Mei et al. [36]
	Zhang et al. [59]	Liu et al. [33]	Wu et al. [51]
	Jiang et al. [22]	Johnson et al. [23]	Mei et al. [35]
	Shi et al. [40]		Chen et al. [4]
	Zhang et al. [56]		Liang et al. [28]
	Wang et al. [46]		Conde et al. [8]
			Chen et al. [6]
Stereo image SR	Fixed disparity	Horizontal disparity	Irregular disparity
	Jeon et al. [21]	Wang et al. [43]	Zhu et al. [61]
		Wang et al. [48]	Chen et al. [3]
		Yan et al. [52]	
		Lei et al. [26]	
		Dai et al. [9]	
		Zhang et al. [60]	
		Chu et al. [7]	
		Lin et al. [30]	
		Liu et al. [31]	

network using known I_{HR} and I_{LR} image pairs by minimizing the objective function

$$\hat{\theta} = \arg\min_{\theta} \mathcal{L}(I_{SR}, I_{HR}) + \lambda\Phi(\theta), \tag{3}$$

where $\mathcal{L}$ represents the loss function (e.g., L1 loss, MSE loss, perceptual loss [23], adversarial loss [25], etc.) which measures the difference between the network output I_{SR} and ground-truth I_{HR}; $\Phi(\theta)$ is a regularization term used to avoid overfitting; and λ is a tradeoff parameter.

In stereo image SR, a pair of degraded LR images are captured simultaneously from different views according to (1). We denote the degraded left and right LR image by I^l_{LR} and I^r_{LR}, respectively. Accordingly, the stereo image SR network $\mathcal{N}$ with parameters δ takes as input a pair of LR images and generates their SR counterparts, which is formulated as

$$I^l_{SR}, I^r_{SR} = \mathcal{N}(I^l_{LR}, I^r_{LR}; \delta). \tag{4}$$

The network is trained by using known paired stereo images, which is formulated as

$$\hat{\delta} = \arg\min_{\delta} \mathcal{L}(I^l_{SR}, I^l_{HR}) + \mathcal{L}(I^r_{SR}, I^r_{HR}) + \lambda\Phi(\delta), \tag{5}$$

where, I^l_{HR} and I^r_{HR} denote the left and right HR images without distortion, respectively. Generally, different variants of gradient descent can be utilized to solve (5), such as stochastic gradient descent with momentum [63], Adagrad [64], Adam [65], and AdamW [34].

In the next section, we describe the details of $\mathcal{N}$ in terms of the overall network architecture and the key network components. We also describe the training method used to find the optimal network parameters $\hat{\delta}$ in (5).

4 Method

4.1 Network architecture

The overall network architecture of the proposed SSRT model is illustrated in Fig. 1. The network first maps the left and right input images into a high-dimensional feature space via a 3×3 convolution layer. Given the inputs I_{LR}^{l}, $I_{LR}^{r} \in \mathbb{R}^{H \times W \times 3}$, this step can be formulated as

$$\begin{aligned} I_0^l &= \mathrm{Conv}(I_{LR}^l), \\ I_0^r &= \mathrm{Conv}(I_{LR}^r), \end{aligned} \tag{6}$$

where I_0^l, $I_0^r \in \mathbb{R}^{H \times W \times C}$ are the obtained shallow features. Next, multi-kernel Transformer blocks and cross-merging blocks are cascaded alternatively to extract refined image details, which is formulated as

$$\begin{aligned} F_i^{l/r} &= \mathrm{MKTB}(I_i^{l/r}), \quad i = 0, 1, ..., N-1, \\ I_{j+1}^l, I_{j+1}^r &= \mathrm{CMB}(F_j^l, F_j^r), \quad j = 0, 1, ..., N-2, \end{aligned} \tag{7}$$

where $\mathrm{MKTB}(\cdot)$ and $\mathrm{CMB}(\cdot, \cdot)$ represent application of the multi-kernel Transformer block (MKTB) and the cross-merging block (CMB), respectively; $F_i^{l/r} \in \mathbb{R}^{H \times W \times C}$ are the intra-view features captured by MKTB; $I_i^{l/r} \in \mathbb{R}^{H \times W \times C}$ are the refined features fused by CMB; and N denotes the total number of MKTBs. The outputs of the last MKTBs are then fed into upsampling modules consisting of convolution layers and pixel shuffle layers [40]. Finally, the outputs of the upsampling modules are added to bicubic-interpolated LR images to generate the reconstructed image. This process can be formulated as

$$I_{SR}^{l/r} = \mathrm{Upsample}(F_{N-1}^{l/r}) + \mathrm{Bicubic}(I_{LR}^{l/r}), \tag{8}$$

where $\mathrm{Upsample}(\cdot)$ and $\mathrm{Bicubic}(\cdot)$ denote, respectively, application of the upsampling module and the bicubic interpolation operation; $I_{SR}^{l/r} \in \mathbb{R}^{sH \times sW \times 3}$ denotes the super-resolved HR image with an upscale factor of s. In SSRT, the two parallel branches for the two views share the same network parameters such that the model complexity can be significantly reduced and cross-view attention can be computed within an identical feature space. Overall, the forward propagation process of SSRT is summarized in Algorithm 1. We provide details of MKTB and CMB in the following subsections.

4.2 Multi-Kernel transformer block

In image restoration or single image SR, it is often important to extract intra-view features. Existing methods use either local spatial self-attention which functions as an adaptive spatial convolution or transposed attention which functions as a point-wise convolution, but not both. In this paper, we present an MKTB by using both neighborhood attention and transposed attention to jointly utilize adaptive large-kernel convolution and point-wise convolution for more effective intra-view feature extraction.

As shown in Fig. 1, the MKTB consists of an NATB and a dual-window TATB. Given a tensor $X \in \mathbb{R}^{H \times W \times C}$, the forward pass of MKTB is formulated as

$$\begin{aligned} Y &= \mathrm{NATB}(X), \\ Z &= \mathrm{TATB}(Y), \end{aligned} \tag{9}$$

Algorithm 1 Pseudo-code of the forward propagation process of SSRT.

Input: $I^l_{LR}, I^r_{LR} \in \mathbb{R}^{H\times W\times 3}$
Output: $I^l_{SR}, I^r_{SR} \in \mathbb{R}^{sH\times sW\times 3}$ ▷ s denotes the scaling factor
$I^l_0 \leftarrow \text{Conv}(I^l_{LR})$
$I^r_0 \leftarrow \text{Conv}(I^r_{LR})$
$i \leftarrow 0$
while $i < N$ **do** ▷ N denotes the number of MKTBs
  $F^l_i \leftarrow \text{MKTB}_i(I^l_i)$
  $F^r_i \leftarrow \text{MKTB}_i(I^r_i)$
  if $i < N-1$ **then** ▷ $N-1$ denotes the number of CMBs
    $I^l_{i+1}, I^r_{i+1} \leftarrow \text{CMB}_i(F^l_i, F^r_i)$
  end if
  $i \leftarrow i+1$
end while
$I^l_{SR} \leftarrow \text{Upsample}(F^l_{N-1}) + \text{Bicubic}(I^l_{LR})$
$I^r_{SR} \leftarrow \text{Upsample}(F^r_{N-1}) + \text{Bicubic}(I^r_{LR})$

where $Y, Z \in \mathbb{R}^{H\times W\times C}$ are the output feature maps of the NATB and the TATB, respectively.

4.2.1 Neighborhood attention transformer block

The NATB consists of an enhanced multi-head neighborhood attention (EMNA) block and a gated depth-wise convolution feed-forward network (GDFN) [54]; LayerNorm (LN) layers [1] and residual connections are employed for both modules to improve the training process. The whole process is formulated as

$$\begin{aligned} \hat{Y} &= \text{EMNA}(\text{LN}(X)) + X, \\ Y &= \text{GDFN}(\text{LN}(\hat{Y})) + \hat{Y}. \end{aligned} \tag{10}$$

As illustrated in Fig. 3, we first project the normalized input tensor to $Q, K, V \in \mathbb{R}^{H\times W\times d}$ by using a point-wise convolution followed by a 3×3 depth-wise convolution. Here, we set $d = 32$, and define attention weights $A_k(i,j) \in \mathbb{R}^{k^2}$ with neighborhood size $k\times k$ for the query location (i,j) as

$$\begin{aligned} A_k(i,j) &= [Q(i,j)^\top K(u,v)] + B, \\ u &\in [i - \lfloor k/2 \rfloor, i + \lfloor k/2 \rfloor], \\ v &\in [j - \lfloor k/2 \rfloor, j + \lfloor k/2 \rfloor], \end{aligned} \tag{11}$$

where $Q(i,j), K(i,j) \in \mathbb{R}^d$ are the corresponding pixel-wise features; and B is a relative positional bias. Then, we extract neighborhood values $V_k(i,j) \in \mathbb{R}^{k^2\times d}$ and obtain the corresponding response $O(i,j) \in \mathbb{R}^d$ at the spatial location (i,j) as

$$O(i,j) = \text{Softmax}(A_k(i,j)^\top/\tau)V_k(i,j), \tag{12}$$

where τ is a learnable temperature factor which controls the magnitude of the attention weights before applying the Softmax operation. Finally, we concatenate the responses of the heads and fuse cross-channel features by using a point-wise convolution to obtain the output of EMNA, which is denoted as $\hat{Y}$.

Next, $\hat{Y}$ is fed into a GDFN after being normalized by LN. As shown in Fig. 4, the normalized input passes through two parallel branches, each of which contains a 1×1 convolution to expand the number of feature channels followed by a 3×3 depth-wise convolution to encode

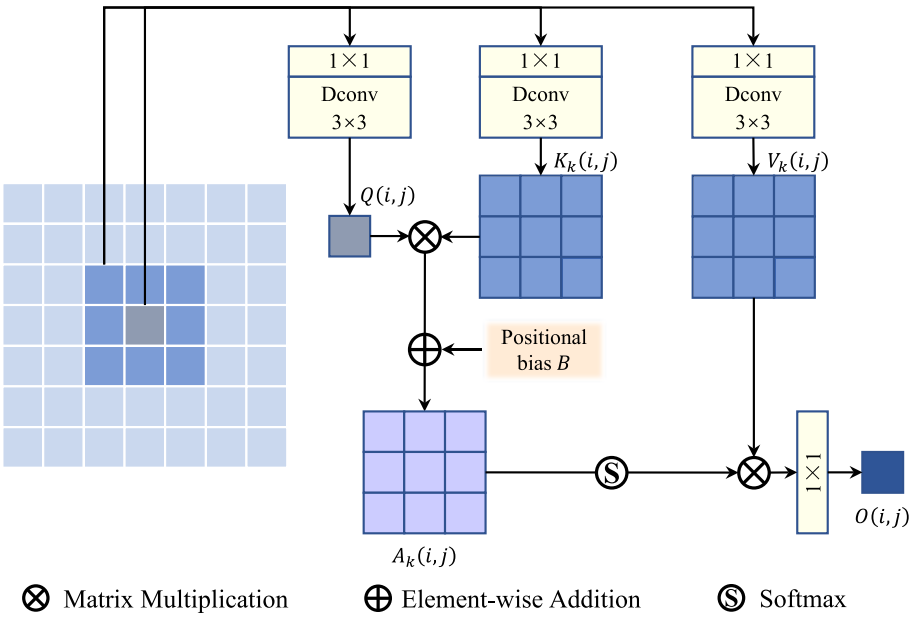

Fig. 3 Network architecture of the enhanced multi-head neighborhood attention mechanism with a neighborhood size of 3×3. Note that the figure shows the attention output for only one of the query positions; the other query positions are processed similarly

the neighboring spatial context; one branch is activated by the GELU non-linearity [16]. Then, the outputs of the element-wise multiplication of the two branches are projected to the original input dimension by using a 1×1 convolution. Overall, the GDFN is formulated as

$$\text{GDFN}(\text{LN}(\hat{Y})) = W_p^0 \text{Gating}(\text{LN}(\hat{Y})), \tag{13}$$

where

$$\text{Gating}(B) = \sigma(W_d^1 W_p^1 B) \odot W_d^2 W_p^2 B. \tag{14}$$

Here, LN denotes layer normalization; σ denotes the GELU non-linearity; $\odot$ denotes the element-wise multiplication; W_p^i $(i = 0, 1, 2)$ denotes the weight of the i-th 1×1 point-wise convolution layer; and W_d^j $(j = 1, 2)$ denotes the weight of the j-th 3×3 depth-wise

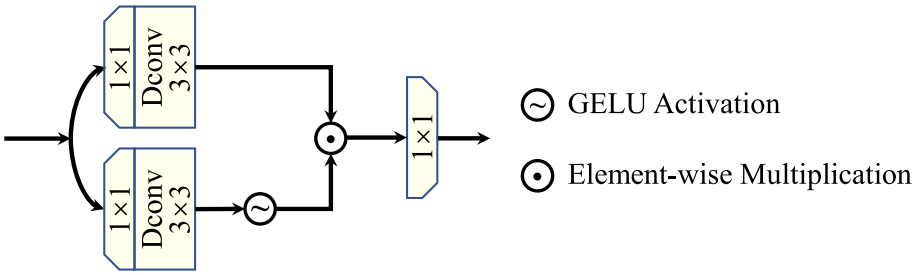

Fig. 4 An architecture of the gated depth-wise convolution feed-forward network

convolution layer. Finally, the output of NATB is obtained by adding $\hat{Y}$ to the output of GDFN.

4.2.2 Transposed attention transformer block

Conventional transposed attention [54] performs adaptive 1×1 convolution over all spatial locations using a consistent attention kernel of shape $\mathbb{R}^{C \times C}$. In this paper, we incorporate transposed attention with a dual-window mechanism which consists of fixed and shifted window-partitioning strategies. In transposed attention as shown in Fig. 5, we observe that the multi-head mechanism divides the global cross-channel receptive field into individual heads of fewer feature channels to reduce the computational cost, and the window partition allows a channel to attend to other channels using different attention kernels in different image regions. Thus, the advantages of our proposed dual-window mechanism reflect not only on the improved model capacity in adaptive feature representation, but also the ability to avoid blocking artifacts that could be induced if we were to use a fixed window-partitioning strategy.

By replacing EMNA in NATB with the dual-window multi-head transposed attention (DMTA), we construct the TATB module. The forward propagation of TATB is formulated as

$$\begin{aligned} \hat{Z} &= \mathrm{DMTA}(\mathrm{LN}(Y)) + Y, \\ Z &= \mathrm{GDFN}(\mathrm{LN}(\hat{Z})) + \hat{Z}. \end{aligned} \tag{15}$$

The essential computational module of DMTA is the window-based transposed attention as shown in Fig. 6. In DMTA, the input is first projected to $Q_i, K_i, V_i \in \mathbb{R}^{H \times W \times \frac{C}{h}}$ using three transforms respectively after being normalized by LN. Here, i denotes the i-th head. Note that each of the projection transforms consists of a 1×1 convolution layer and a 3×3 depth-wise convolution layer, and operates independently for each head. Then, Q_i, K_i, V_i are split into $\frac{HW}{w^2}$ groups of windows $q_i^g, k_i^g, v_i^g \in \mathbb{R}^{w^2 \times \frac{C}{h}}$, where g denotes the g-th group

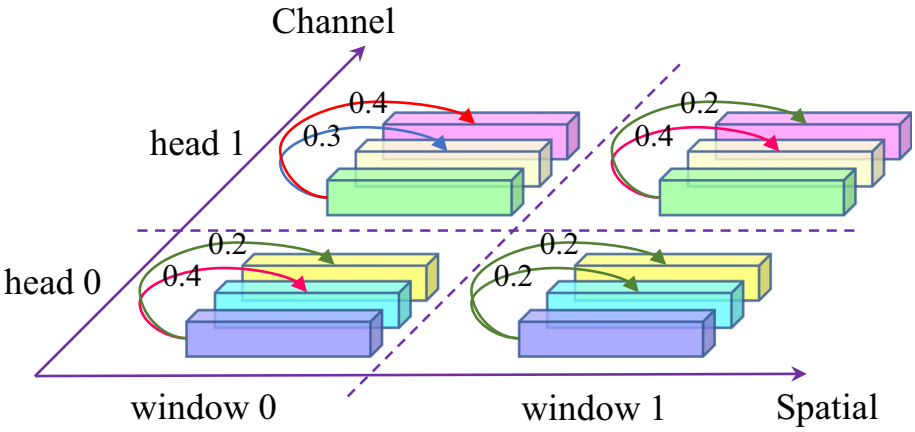

Fig. 5 An illustration of multi-head transposed attention with window-partitioning. The multi-head is used to divide the global cross-channel receptive field into individual heads of small feature channels, and the window partition is utilized to provide extra representation subspaces

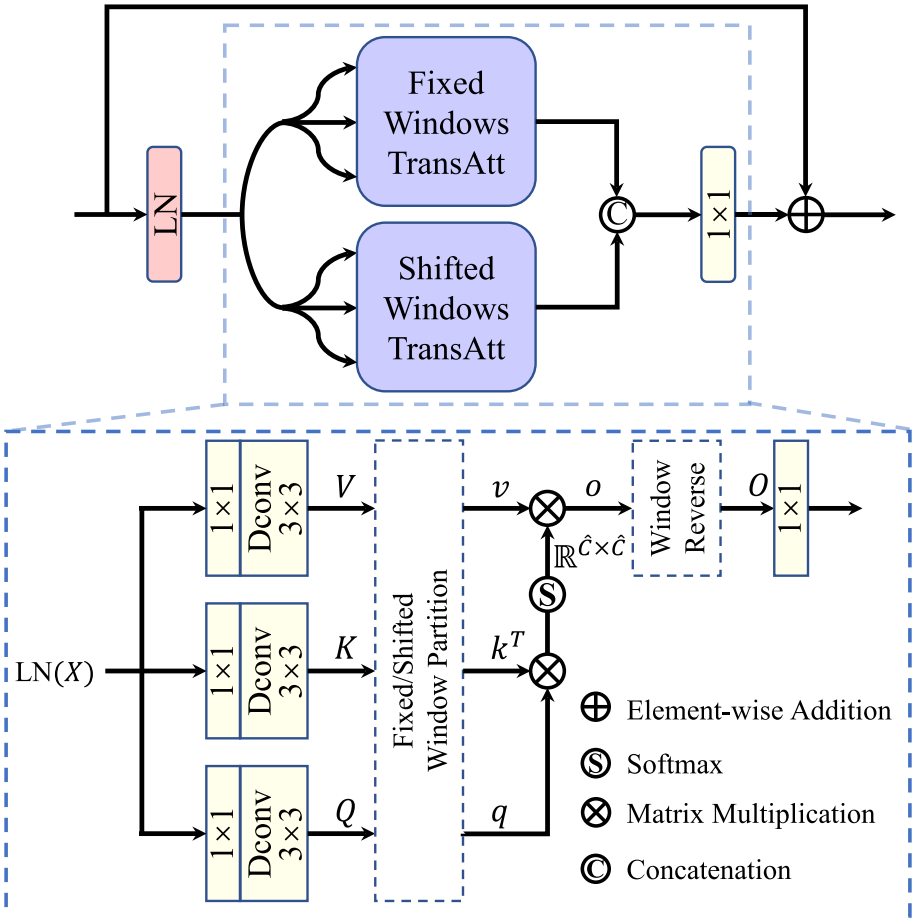

Fig. 6 Network architecture of the dual-window transposed attention mechanism. In DMTA, half of the heads use a fixed-window partition and the other half use a shifted-window partition

window. The transposed attention for each window is computed as

$$o_i^g = v_i^g \text{Softmax}(k_i^{g\top} q_i^g / \tau), \tag{16}$$

where τ is a learnable scaling factor that controls the magnitude of the dot product of q_i and k_i. The reversed window partition is conducted for all o_i^g along the g-axis to achieve the i-th head output $O_i \in \mathbb{R}^{H \times W \times \frac{C}{h}}$. Finally, the output of DMTA is given by

$$\text{DMTA}(\text{LN}(Y)) = W\text{Concat}(O_1, ..., O_h), \tag{17}$$

where W denotes a 1×1 convolution layer used to aggregate the pixel-wise cross-channel context from all heads. This fixed window-partitioning mechanism has the potential to introduce blocking artifacts to the reconstructed images. Thus in DMTA, half of the heads use a fixed window partition, and the other half use a shifted window partition which displaces the windows by $\lfloor \frac{w}{2} \rfloor$ pixels in both directions (horizontal and vertical) before the window partitioning is performed.

4.3 Cross-merging block

In stereo image SR, it is critical to capture the cross dependencies between the image pairs. In existing methods, these dependencies are either limited to the horizontal epipolar or restricted to a fixed receptive field of the network. Hence, we propose a more flexible cross-merging block to better capture the cross dependencies between the two views.

The network architecture of the cross-merging block is illustrated in Fig. 7. Specifically, two transform layers (M-layer and V-layer) are first employed to map the input (X^l and X^R) to the matching and transferring spaces, respectively. Then, the output of the M-layer is bilinearly downsampled before patch-wise attention is applied in order to reduce the computational cost. Here, each transform consists of a 1×1 convolution layer followed by a 3×3 depth-wise convolution layer. In the LR space, the inputs M^l, $M^r \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times d}$ are first unfolded to 3×3-pixel patches m^l_{ij} and m^r_{ij}, where the subscripts i, j denote that the patch is centered at the i-th row and j-th column. Then, we search the right view for the most similar patch to the left-view patch m^l_{ij}, and record the position $\mathrm{P}^{\mathrm{r}\to\mathrm{l}}$ and similarity $\mathrm{S}^{\mathrm{r}\to\mathrm{l}}$ which are given by

$$s^{r\to l}_{ij} = \max_{u,v} \left\langle \frac{m^l_{ij}}{\left\| m^l_{ij} \right\|}, \frac{m^r_{uv}}{\left\| m^r_{uv} \right\|} \right\rangle, \tag{18}$$

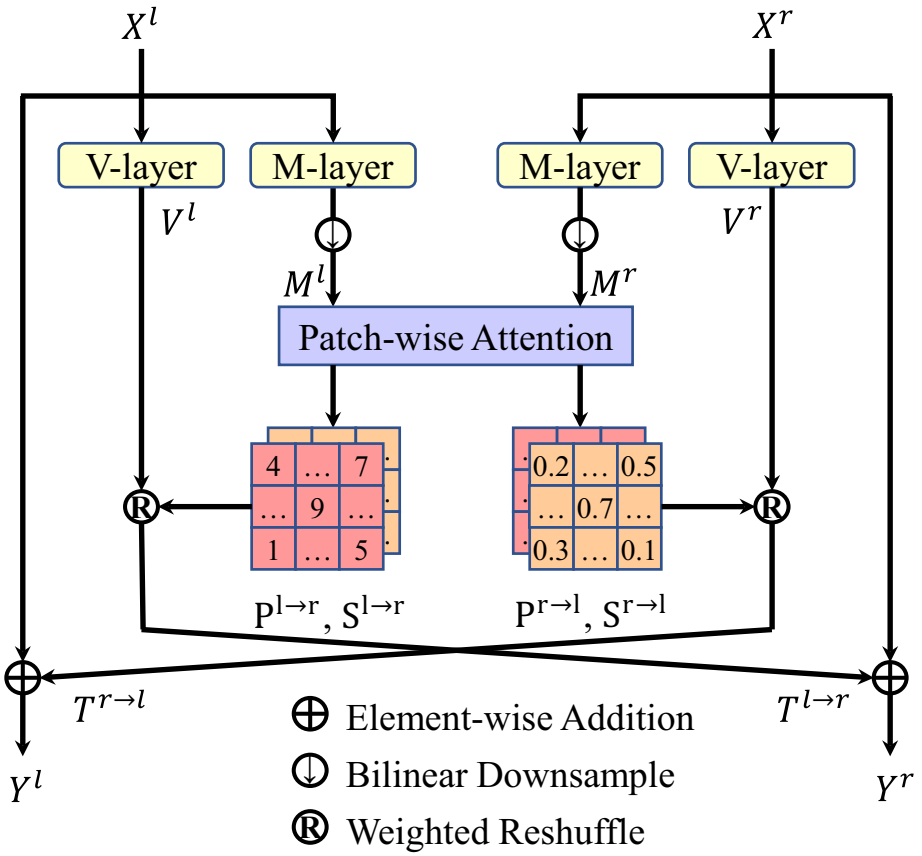

Fig. 7 Network architecture of the cross-merging block

and

$$p_{ij}^{r\to l} = \arg\max_{u,v} \left\langle \frac{m_{ij}^{l}}{\left\| m_{ij}^{l} \right\|}, \frac{m_{uv}^{r}}{\left\| m_{uv}^{r} \right\|} \right\rangle, \tag{19}$$

where $u \in [i-n, i+n]$ and $v \in [0, W-1]$ denote the coordinates of the patch center; n denotes the search increment; $s_{ij}^{r\to l}$ and $p_{ij}^{r\to l}$ denote respectively the element of $\mathrm{S}^{\mathrm{r}\to\mathrm{l}}$ and $\mathrm{P}^{\mathrm{r}\to\mathrm{l}}$ in the i-th row and j-th column. Note that we address the boundary overflow of u by using a cyclic shift. Accordingly, the similar patch is transferred based on $s_{ij}^{r\to l}$ and $p_{ij}^{r\to l}$ via

$$t_{ij}^{r\to l} = s_{ij}^{r\to l} \cdot v_{\hat{p}_{ij}^{r\to l}}^{r}, \tag{20}$$

where v^r denotes the unfolded 6×6-pixel patches derived from the output of the V-layer applied to the right view; $t_{ij}^{r\to l}$ denotes the patch at the coordinate (i, j) transferred from the right view to the left view; and $\hat{p}_{ij}^{r\to l}$ denotes the transformed position from the matching space to the transferring space. According to the match strategy, we can dynamically tune the parameter n during the testing stage to address stereo image pairs with different epipolar constraints, instead of training a new network from scratch. Finally, the transferred features are added to the original input:

$$Y^l = X^l + T^{r\to l}, \tag{21}$$

where $T^{r\to l}$ is obtained by folding all $t_{ij}^{r\to l}$ and by dividing by the number of times an overlap occurred at the corresponding position. The output features corresponding to the right view are calculated in a similar fashion.

The total computational complexity of proposed patch-wise attention is $\mathcal{O}(\frac{H}{2} \cdot \frac{W}{2} \cdot (2n+1) \cdot \frac{W}{2} \cdot 9C + \frac{H}{2} \cdot \frac{W}{2} \cdot (2n+1) \cdot \frac{W}{2}) = \mathcal{O}(\frac{9(2n+1)}{8} HW^2C + \frac{2n+1}{8} HW^2)$, and the vertical receptive field size of CMB is $4n+1$. In comparison, in normal pixel-wise attention, the $4n+1$ vertical search region results in an $\mathcal{O}(2(4n+1)HW^2C)$ computational cost, which is about four times that of our proposed method.

4.4 Loss function

The L_1 loss that measures the pixel-wise absolute difference between the restored and ground-truth images has been widely used in various image restoration tasks, and was demonstrated to provide better convergence and performance than L_2 loss [29]. Thus, we trained our network by minimizing the L_1 loss function, which is given by

$$\mathcal{L} = \left\| I_{SR}^{l} - I_{HR}^{l} \right\|_1 + \left\| I_{SR}^{r} - I_{HR}^{r} \right\|_1, \tag{22}$$

where $I_{SR}^{l/r}$ denotes the reconstructed left/right HR view, and $I_{HR}^{l/r}$ denotes the corresponding ground-truth image.

5 Experiment

In this section, we first describe the datasets and metrics, and then provide details of our model implementation. Next, we conduct quantitative and qualitative evaluations to compare our model with other state-of-the-art methods. Finally, we perform ablation studies to verify the utilities of the key network components.

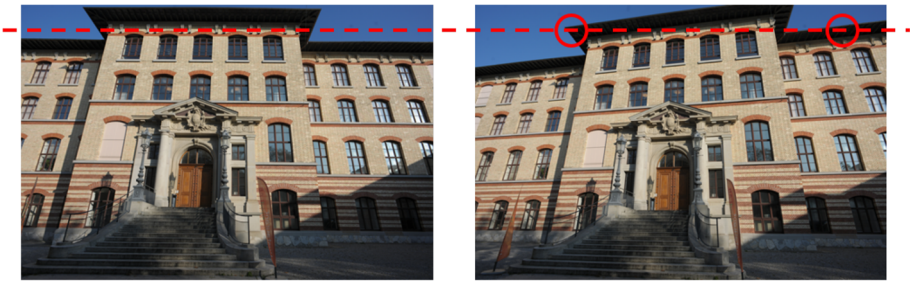

Fig. 8 A stereo image pair from the ETH3D dataset. Note that the red circles mark examples of objects that are not located on the same horizontal line in the two views

5.1 Datasets and metrics

Following [7, 48], our training dataset consists of 800 stereo image pairs from the training set of the Flickr1024 dataset [47] and 60 downsampled[1] (by a factor of 2 in each direction) image pairs from the Middlebury dataset [38]. To evaluate the performance of our approach, five public benchmark datasets were used for testing, which include 5 image pairs from Middlebury [38], 20 image pairs from KITTI 2012 [14], 20 image pairs from KITTI 2015 [37], 112 image pairs from the test set of Flickr1024 [47], and 10 image pairs from ETH3D [39] with irregular epipolar constraints (as shown in Fig. 8). Note that the pristine images from ETH3D were downsampled by a factor of 6 in each direction to generate the target HR images. To obtain the LR training/testing images, we downsampled the HR images by using the corresponding scale factors. During training, the LR images were cropped to 48×96 pixels with a stride of 20 pixels. In total, we generated 273,495 and 41,236 image pairs for training the $2\times$ and $4\times$ SR models, respectively. The SR performance was evaluated based on Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM) [49]. PSNR measures the average pixel-wise difference between two images, and the SSIM index operates based on similarity measurements of three elements: luminance, contrast, and structure. Following [7, 9, 48], both criteria were calculated on the left image with 64 pixels on the left boundary being cropped (denoted in Table 2 by $Left$) and on image pairs without cropping (denoted in Table 2 by $(Left + Right)/2$).

5.2 Implementation details

In our implementation, we set the following hyper-parameters: $C = 64$, $d = 32$, $k = 9$, $h = 4$, and $w = 16$. The number of multi-kernel Transformer blocks and cross-merging blocks was set to 12 and 11, respectively. We set $n = 4$ when testing on ETH3D, and 0 for training and testing on other datasets which contain only horizontal offsets. Horizontal/vertical flip, random channel shuffle, and mixup [55] were used as data augmentation strategies to further improve the network generalization, and the batch size was set to 32. Our model was trained with the AdamW optimizer [34] using $\beta_1 = 0.9$ and $\beta_2 = 0.9$ for 2×10^5 iterations. The initial learning rate was set to 2×10^{-3} and the weight decay was set to 10^{-4}. We used cosine annealing [20] to gradually decrease the learning rate until 10^{-6}. Furthermore, following [7], we used skip-init [11] and stochastic depth [19] with a factor of 0.1 to stabilize the training process as well as to avoid overfitting.

[1] In this paper, the bicubic method provided in MATLAB was used to downsample the images.

Table 2 Quantitative results of the proposed SSRT model vs. competing single/stereo image SR methods on various stereo image datasets measured in terms of PSNR and SSIM

Method	Scale	# Param	$Left$			$(Left + Right)/2$				
			KITTI2012	KITTI2015	Middlebury	KITTI2012	KITTI2015	Middlebury	Flickr1024	ETH3D
Bicubic	2×	/	28.64/0.8851	28.01/0.8863	30.67/0.9013	28.71/0.8884	28.82/0.9020	30.81/0.9024	25.09/0.8243	32.16/0.9100
EDSR [29]	2×	38.63M	30.87/0.9199	29.97/0.9232	34.93/0.9491	31.00/0.9229	30.77/0.9337	35.03/0.9494	28.69/0.9097	36.62/0.9517
RDN [59]	2×	21.99M	30.85/0.9197	29.93/0.9225	34.94/0.9490	30.98/0.9227	30.73/0.9331	35.04/0.9494	28.67/0.9094	36.58/0.9517
RCAN [58]	2×	15.31M	30.91/0.9202	30.00/0.9232	34.87/0.9483	31.05/0.9232	30.79/0.9338	34.97/0.9487	28.65/0.9089	36.62/0.9515
SwinIR [28]	2×	11.75M	31.03/0.9221	30.13/0.9253	35.11/0.9505	31.17/0.9251	30.93/0.9356	35.21/0.9510	28.91/0.9121	36.95/0.9531
StereoSR [21]	2×	1.08M	29.43/0.9037	28.54/0.9035	33.17/0.9341	−/−	−/−	−/−	−/−	−/−
PASSRnet [43]	2×	1.37M	30.82/0.9182	29.88/0.9204	34.47/0.9450	30.95/0.9212	30.66/0.9311	34.56/0.9451	28.52/0.9070	36.10/0.9482
IMSSRnet [26]	2×	6.84M	30.90/−	29.97/−	34.66/−	30.92/−	30.66/−	34.67/−	−/−	−/−
iPASSR [48]	2×	1.38M	31.00/0.9210	30.04/0.9235	34.50/0.9455	31.14/0.9240	30.84/0.9341	34.59/0.9456	28.63/0.9106	36.15/0.9493
C2FNet [31]	2×	1.16M	31.04/0.9220	30.10/0.9250	35.07/0.9500	31.19/0.9250	30.90/0.9350	35.12/0.9500	29.06/0.9150	−/−
NAFSSR-S [7]	2×	1.54M	31.24/0.9241	30.29/0.9270	35.24/0.9519	31.38/0.9270	31.09/0.9372	35.31/0.9518	29.20/0.9164	36.77/0.9518
SSRT (Ours)	2×	1.55M	**31.37/0.9263**	**30.35/0.9288**	**35.83/0.9566**	**31.52/0.9293**	**31.15/0.9386**	**35.88/0.9565**	**29.56/0.9226**	**37.57/0.9567**
Bicubic	4×	/	24.64/0.7334	23.90/0.7099	26.39/0.7564	24.70/0.7397	24.51/0.7368	26.52/0.7584	21.88/0.6320	28.07/0.7874
EDSR [29]	4×	38.90M	26.28/0.7943	25.40/0.7801	29.17/0.8374	26.37/0.8006	26.06/0.8031	29.25/0.8389	23.47/0.7289	30.66/0.8614
RDN [59]	4×	22.04M	26.25/0.7943	25.39/0.7804	29.17/0.8378	26.34/0.8006	26.06/0.8035	29.30/0.8396	23.48/0.7302	30.68/0.8619
RCAN [58]	4×	15.36M	26.38/0.7958	25.55/0.7826	29.22/0.8372	26.46/0.8020	26.24/0.8060	29.32/0.8390	23.49/0.7292	30.75/0.8623
SwinIR [28]	4×	11.90M	26.46/0.7998	25.62/0.7870	29.13/0.8405	26.55/0.8061	26.34/0.8103	29.25/0.8420	23.54/0.7319	30.77/0.8659
StereoSR [21]	4×	1.08M	24.50/0.7487	23.68/0.7257	27.70/0.8022	−/−	−/−	−/−	−/−	−/−
PASSRnet [43]	4×	1.42M	26.26/0.7909	25.42/0.7761	28.62/0.8221	26.35/0.7972	26.08/0.7993	28.71/0.8224	23.21/0.7173	30.09/0.8445
IMSSRnet [26]	4×	6.89M	26.44/−	25.59/−	29.02/−	26.43/−	26.20/−	29.02/−	−/−	−/−
iPASSR [48]	4×	1.43M	26.48/0.7984	25.63/0.7840	29.09/0.8353	26.58/0.8045	26.33/0.8076	29.18/0.8358	23.45/0.7291	30.54/0.8573

Table 2 continued

Method	Scale	# Param	$Left$			$(Left + Right)/2$				
			KITTI2012	KITTI2015	Middlebury	KITTI2012	KITTI2015	Middlebury	Flickr1024	ETH3D
SSRDE-FNet [9]	4×	2.26M	26.61/0.8022	25.75/0.7892	29.29/0.8407	26.70/0.8083	26.48/0.8126	29.36/0.8403	23.56/0.7348	30.65/0.8599
C2FNet [31]	4×	1.16M	26.58/0.8010	25.70/0.7880	29.26/0.8370	26.66/0.8070	26.40/0.8110	29.32/0.8370	23.59/0.7350	−/−
NAFSSR-S [7]	4×	1.56M	**26.84**/0.8091	**26.04**/0.7982	29.62/0.8486	**26.93**/0.8150	**26.76**/0.8208	29.72/0.8494	23.88/0.7472	31.00/0.8667
SSRT (Ours)	4×	1.70M	26.83/**0.8107**	25.95/**0.7987**	**29.75/0.8543**	26.92/**0.8168**	26.68/**0.8216**	**29.87/0.8554**	**23.95/0.7535**	**31.37/0.8767**

Here, the PSNR/SSIM values achieved on both the left images (i.e., $Left$) and the stereo image pairs (i.e., $(Left + Right)/2$) are presented. Note that # Param denotes the number of parameters of the networks. The best results are in **bold** and the second best results underlined

All experiments were conducted on a remote server with an Intel Xeon Silver 4214 CPU and an NVIDIA GeForce RTX 3090 GPU. The operating system was Debian GNU/Linux 11, and the CUDA and CuDNN versions were 11.3 and 8302, respectively. Our model was implemented by using PyTorch 1.12.0. A single GPU was used for training, which took about six days to finish.

5.3 Comparisons with other methods

We compared SSRT with several state-of-the-art SISR and stereo image SR methods. The five SISR methods were bicubic interpolation, EDSR [29], RDN [59], RCAN [58], and SwinIR [28]. The seven stereo image SR methods were StereoSR [21], PASSRnet [43], iPASSR [48], IMSSRnet [26], SSRDE-FNet [9], C2FNet [31], and NAFSSR-S [7].

1) Quantitative Results. The quantitative comparison results tested on the five datasets with respect to PSNR and SSIM are presented in Table 2. Note that the results of IMSSRnet [26] and C2FNet [31] were directly obtained from the original papers. Larger values indicate better image quality. Also included in Table 2 are the network parameter numbers for each SR model. As can be observed, our approach achieves either the best or highly competitive performance as compared with other SR methods by using a comparable number of network parameters. Specifically, for the $2\times$ SR task, SSRT performs the best on all datasets. The average PSNR values achieved by SSRT surpass NAFSSR-S by 0.14 dB, 0.06 dB, 0.57 dB, and 0.36 dB on KITTI 2012, KITTI 2015, Middlebury, and Flickr1024, respectively. On ETH3D images with irregular epipolar constraints, SSRT surpass NAFSSR-S by 0.80 dB in terms of PSNR, which demonstrates the great effectiveness of the proposed cross-merging block in coping with vertical pixel offsets. For the $4\times$ SR task, SSRT also demonstrates the best/competitive performance. The average PSNR values achieved by SSRT on Middlebury, Flickr1024, and ETH3D surpass NAFSSR-S by 0.15 dB, 0.07 dB, and 0.37 dB, respectively.

2) Computational Complexity. To investigate the computational complexity of SSRT and other SR methods, stereo image pairs of 256×256-pixel size were used for testing on the $4\times$ SR task. The experiment was conducted on the same remote server as described in Section 5.2. The number of parameters, inference time (averaged over 20 stereo images), and the number of floating point operations (FLOPs) for each method are provided in Table 3. Observe that SSRT maintains an acceptable computational complexity as compared with other methods.

3) Qualitative Results. In this section, we provide visual comparisons of different SR methods applied on sample images or image pairs from the testing datasets. Figures 9, 10, 11, and 12 show the $4\times$ SR results of different SR algorithms tested on a sample image or image pairs from the five datasets, among which ETH3D contains irregular pixel offsets and the other four contain horizontal disparity only. The corresponding PSNR and SSIM values are presented at the bottom of each image. As can be observed, the images produced by our method generally display sharper textures and are visually closer to the ground-truth as compared to other SR methods.

4) Real-World Super-Resolution. To investigate the performance of SSRT on real-world images, the test set of Holopix50k [18] which consists of 2,468 stereo images was used for testing. Figure 13 shows the $4\times$ SR results of different SR methods on a real stereo image pair of size 360×640 pixels. Observe that SSRT produces visually better HR images with sharper edges and clearer textures than other SR methods. Note that SSRDE-FNet [9] requires a relatively larger amount of GPU memory (more than 24 GB) to operate, indicating that it is almost impossible to apply the method on most consumer-level GPUs.

Table 3 Computational complexity analysis of different image SR methods tested on stereo image pairs with 256×256 pixel size in 4× SR task

Method	EDSR [29]	RDN [59]	RCAN [58]	SwinIR [28]	PASSRnet [43]	iPASSR [48]	SSRDE-Fnet [9]	NAFSSR-S [7]	SSRT (ours)
# Param	38.90M	22.04M	15.36M	11.90M	1.42M	1.43M	2.26M	1.56M	1.70M
Runtime (s)	0.376	0.326	0.325	0.662	0.041	0.043	0.584	0.219	0.321
FLOPs (G)	2606.2	1445.6	1000.0	808.7	163.0	189.5	1709.2	163.3	278.2

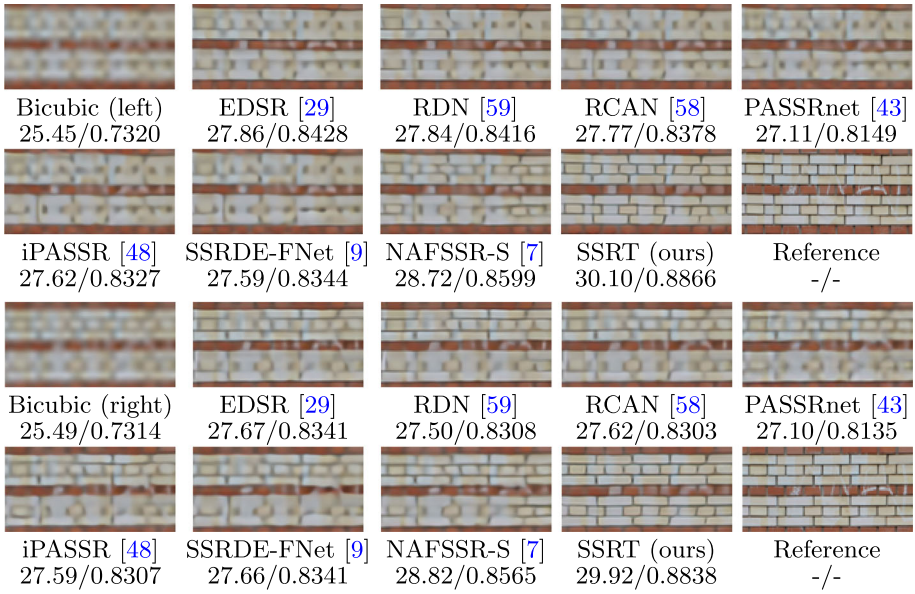

Fig. 9 Visual comparisons of 4× SR results achieved by SSRT and other SR methods on the ETH3D dataset

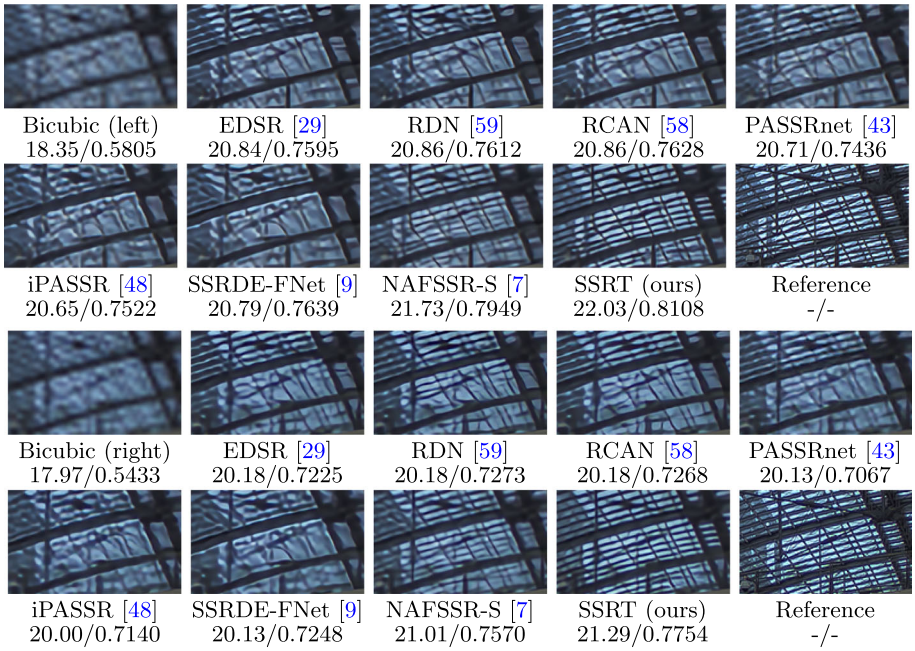

Fig. 10 Visual comparisons of 4× SR results achieved by SSRT and other SR methods on the Flickr1024 dataset

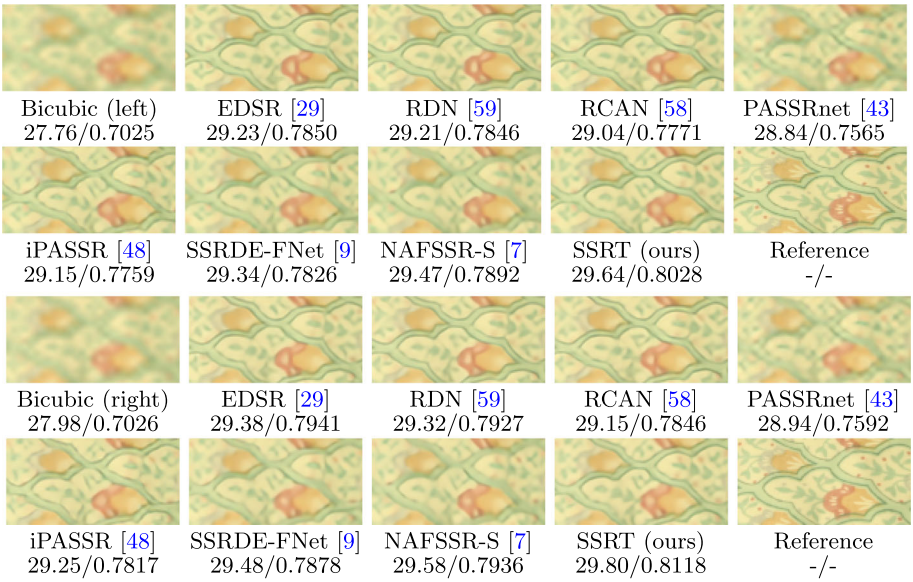

Fig. 11 Visual comparisons of 4× SR results achieved by SSRT and other SR methods on the Middlebury dataset

5.4 Ablation study

We also performed an ablation study to analyze the contributions of the different modules and hyper-parameter settings towards the overall SR performance. Specifically, we used 24 GDFN [54] blocks as the baseline and set the upscale factor to 4. It is important to note that due to the limited computational resources, for Tables 4, 6, and 8, we trained the ablation-study models for 2×10^5 iterations using a batch size of 8 and a constant learning rate of 2×10^{-4}. Thus, the results in the three tables are a bit different from those in Table 2. For Table 5, we trained the models by using the same parameter settings as in Section 5.2. Here, we report the average PSNR values computed on a similarly distributed dataset (i.e., Flickr1024

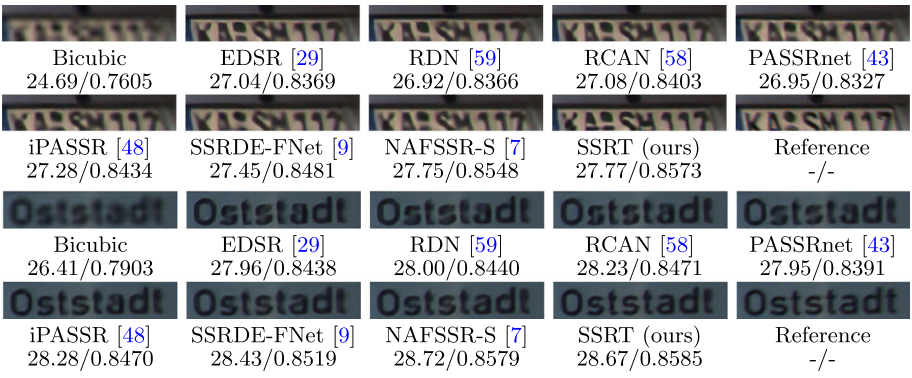

Fig. 12 Visual comparisons of 4× SR results achieved by SSRT and other SR methods on the KITTI2012 (top) and KITTI2015 (bottom) dataset

Fig. 13 Visual comparisons of 4× SR results achieved by SSRT and other SR methods on the real-world stereo images of Holopix50k dataset

Table 4 Average PSNR values tested on the Flickr1024 validation set and the ETH3D dataset by using different transposed attention blocks, window sizes (ws), and head numbers

Method	ws	# heads	# Param	Flickr1024	ETH3D
Baseline	–	–	1.18M	23.29	30.49
Global	–	1	1.63M	23.34	30.65
transposed	–	2	1.63M	23.33	30.67
attention [54]	–	4	1.63M	23.34	30.67
Fixed-window	8	1	1.63M	23.41	30.66
transposed	8	2	1.63M	23.40	30.68
Attention	8	4	1.63M	23.41	30.69
	8	2	1.63M	23.43	30.72
Dual-window	8	4	1.63M	23.43	30.73
transposed	4	4	1.63M	23.38	30.62
attention	16	4	1.63M	23.45	30.81
	24	4	1.63M	23.44	30.83
	48	4	1.63M	23.42	30.81

Table 5 Average PSNR values of SSRT tested on the Flickr1024 and ETH3D datasets by using different transformer blocks

MKTB	# Param	Flickr1024	ETH3D
NATB + NATB	1.71M	23.84	31.21
TATB + TATB	1.70M	23.85	31.17
NATB + TATB	1.70M	23.88	31.27

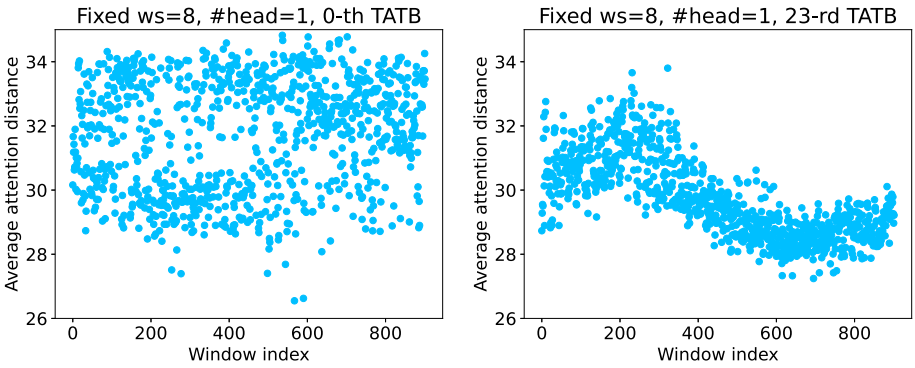

Fig. 14 The average attention distance computed between a query channel and all the key channels within one head in the 0-th and 23-rd TATBs for different windows

validation set) and a dissimilarly distributed dataset (i.e., ETH3D dataset) to quantify the SR performance.

1) Transposed Attention Methods and Head Number. As mentioned previously, the proposed DMTA is an essential element in providing multiple adaptive kernels to improve the model performance. For demonstration, we first incorporated within the baseline model a single-head spatially global transposed attention block [54], and then replaced it with a fixed-window transposed attention block. The result is shown in Table 4. We observe that the global transposed attention [54] obtains a 0.05 dB improvement in terms of PSNR as compared with the baseline, and the fixed-window transposed attention further achieves a 0.07 dB improvement as compared with the global transposed attention. This indicates that the ability to fit similarly distributed data can be improved without introducing additional network parameters and computational cost. Figure 14 shows the average attention distance between a query channel and all the key channels within one head in the TATB for different windows, where the horizontal axis represents the different windows and the vertical axis represents the corresponding attention distance. Note that the average attention distance is computed as the weighted sum of all of the key indices where the weight is the cosine similarity between the key and the query. Observe that different windows attempt to attend to different channels by using different attention weights. Since traditional transposed attention [54] restrains the representation space into a single global window, resulting in a sub-optimal solution is obtained resulting in performance degradation. Next, we increased the head number and used the dual-window mechanism. We can see from Table 4 that the multi-head contributes less to the performance of the global and fixed-window transposed attention. However, for the dual-window approach, the multi-head can not only save computation, but it also further improves the model robustness.

Table 6 Average PSNR values of SSRT tested on the Flickr1024 and ETH3D datasets by using different numbers of CMB

# CMB ($n = 0$)	0	1	2	3	5	11
# Param	1.63M	1.63M	1.64M	1.65M	1.66M	1.70M
Flickr1024	23.45	23.56	23.59	23.59	23.60	23.62
ETH3D	30.81	30.79	30.83	30.82	30.81	30.84

Table 7 Average PSNR values of SSRT tested on the Flickr1024 and ETH3D datasets by using different numbers of MKTBs

# MKTB/CMB	4/3	6/5	8/7	10/9	12/11
# Param	0.76M	1.00M	1.23M	1.47M	1.70M
Flickr1024	23.47	23.56	23.60	23.62	23.63
ETH3D	30.66	30.73	30.85	30.84	30.90

2) Window Size of Transposed Attention. We tested different window sizes to explore the potential of DMTA; the results are shown in Table 4. As can be observed, either a large or small window size will decrease the performance, which is likely attributable to the fact that a large window size will reduce the number of adaptive attention kernels, and a small window size has insufficient query information resulting in an unsatisfactory attention matching. Thus, in this work, we set the window size to 16 to achieve the maximum model performance.

3) Different Transformer Block Combinations. Since the transposed attention can be viewed as an adaptive point-wise convolution, the spatial receptive field of the network can be limited if only the TATB is used in the MKTB. Thus, to investigate the effectiveness of different attention mechanisms, we designed four different combinations for the MKTB: (1) NATB + NATB, (2) TATB + TATB, and (3) NATB + TATB. The results are presented in Table 5. As can be observed, either an excessive increase of the spatial receptive field (NATB + NATB) or a small receptive field (TATB + TATB) will lead to sub-optimal performance. Thus, a combination of a NATB followed by a TATB was adopted in the MKTB.

4) Number of Cross-Merging Blocks. We tested SSRT by using different numbers of CMBs to investigate its impact on the overall performance; the results are shown in Table 6. We can observe from Table 6 that the performance on Flickr1024 improves when the number of CMBs increases because images in the dataset contain only horizontal parallax. However, for ETH3D images which contain irregular epipolar constraints, the performance does not change significantly because the search increment n is set to 0. This fact indicates that taking into account only the horizontal epipolar constraint can be insufficient when processing real stereo images.

5) Number of Multi-kernel Transformer Blocks. We also tested SSRT with different numbers of MKTBs, in which case the number of CMBs was always one less than the number of MKTBs according to the original model design. The testing results are shown in Table 7, from which we can observe that the SR performance is improved when the model depth increases. However, we also observe that deeper models do not always guarantee significantly enhanced performance. Thus, by balancing the performance and complexity of the network, 12 MKTBs were finally adopted in our model.

Table 8 Average PSNR values of SSRT tested on the Flickr1024 and ETH3D datasets by using different search increments and cross attention methods

Search increment n		0	1	2	4	6	8
Flickr1024	Pixel-wise	23.45	23.45	23.45	23.45	23.45	23.45
	Patch-wise	23.59	23.60	23.60	23.59	23.59	23.59
ETH3D	Pixel-wise	30.80	30.80	30.80	30.80	30.80	30.80
	Patch-wise	30.82	30.88	30.89	30.90	30.91	30.91

Note that in this test only three CMBs were considered

6) Different Cross Attention Methods. As mentioned previously, the cross merging block with patch-wise attention plays an important role in capturing the cross dependencies between the two views. To explore the effectiveness of different cross attention methods, we replaced the patch-wise attention with pixel-wise attention which aggregates all spatial positions of the query region via a weighted sum, and gradually increased the search increment n from 0 to 8. Here, only three CMBs were used, and the results are shown in Table 8. As can be observed, the CMB with patch-wise cross attention can effectively capture the vertical parallax as

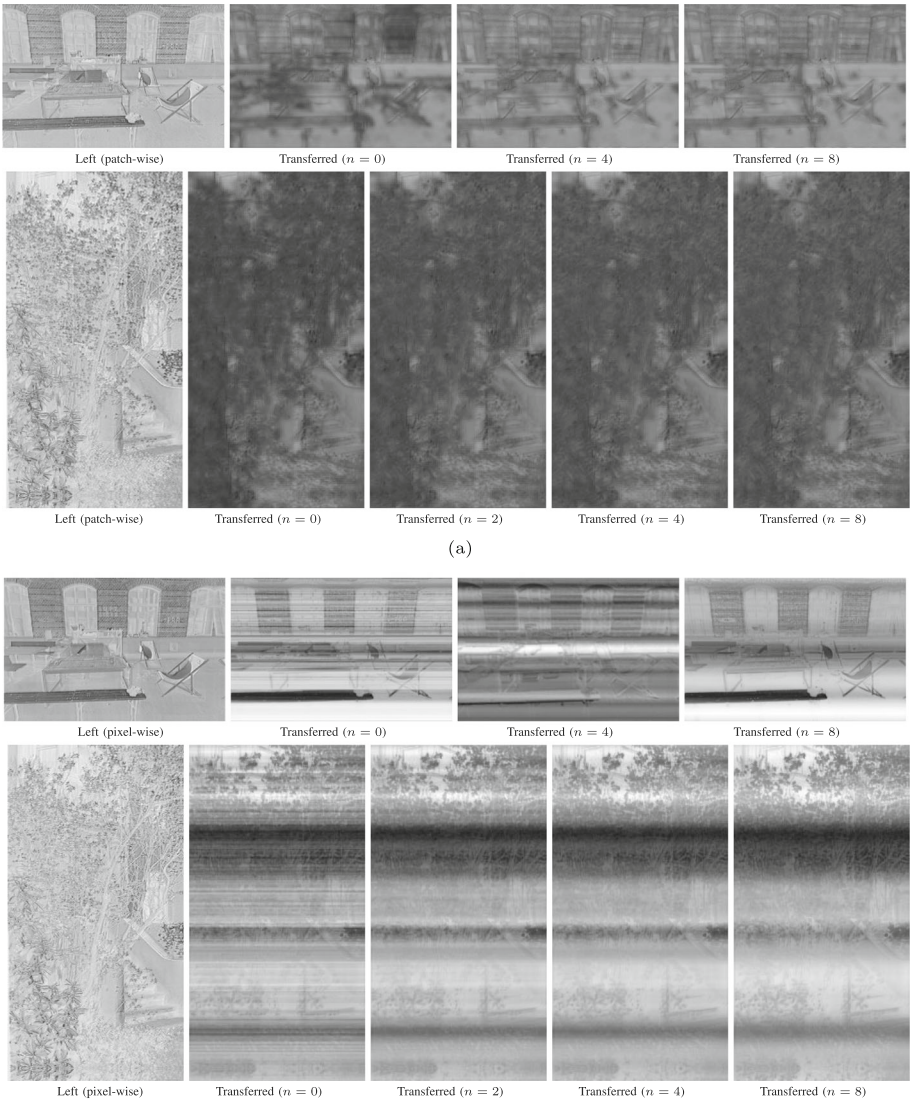

Fig. 15 Visualization of the input feature maps and the corresponding transferred features obtained by using (a) patch-wise cross attention, and (b) pixel-wise cross attention. Note that all transferred features are derived from the 0-th feature map of the first CMB in Fig. 1

n increases while maintaining the ability to take into account a large horizontal disparity. However, for the CMB with pixel-wise cross attention, the performance is inferior to that of the CMB with patch-wise attention, and the performance does not change significantly when n varies. This finding is due to the fact that when patch-wise attention is used, the cross dependencies between the two views can be more effectively captured, with increasing performance as n increases (as shown in Fig. 15a). However, for pixel-wise attention, the transferred features often display noise, and they tend to be smooth for large n values (as shown in Fig. 15b). This might suggest that for cross-view attention, it is unnecessary to aggregate all pixels in the query region, as too many irrelevant points are considered.

6 Conclusion

In this paper, we proposed a multi-kernel Transformer with inductive bias called SSRT for stereo image SR. Specifically, we incorporated a dual-window mechanism within the conventional transposed attention module, and we designed a multi-kernel Transformer block by combining a neighborhood attention Transformer block with a transposed attention Transformer block to balance the receptive field size and the model computational complexity for intra-view feature extraction. The proposed cross-merging block adopts the patch-wise attention mechanism to take into account both vertical and horizontal parallax while maintaining a reasonable computational complexity. Compared with the pixel-wise attention widely used in existing works, the employed patch-wise attention can more accurately capture the cross dependencies between the two views. Experimental results tested on five benchmark datasets demonstrate the superiority of SSRT as compared with other state-of-the-art stereo image SR methods.

Despite the effectiveness of the proposed SSRT model, there are still some limitations. For example, due to the large amount of GPU memory required during training, we adopted the checkpoint strategy provided in PyTorch to save GPU memory. However, this strategy will also recalculate the gradient during the backward propagation which inevitably increases the training time. Thus, future work could focus on developing more efficient intra/cross-view feature extraction mechanisms, as well as developing improved model pruning strategies to save memory. Future work might also focus on designing a unified model that is able to super-resolve LR images at different scale factors.

Funding National Natural Science Foundation of China (61901355, 62271384); Japan Society for the Promotion of Science (22K12085).

Availability of Data and Materials The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of Interest Authors declare that we have no conflict of interest.

References

1. Ba JL, Kiros JR, Hinton GE (2016) Layer normalization. arXiv:1607.06450

2. Chang H, Yeung DY, Xiong Y (2004) Super-resolution through neighbor embedding. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition, 2004. CVPR 2004., IEEE, pp I–I
3. Chen C, Qing C, Xu X et al (2022) Cross parallax attention network for stereo image super-resolution. IEEE Trans Multimed 24:202–216. https://doi.org/10.1109/TMM.2021.3050092
4. Chen H, Wang Y, Guo T et al (2021) Pre-trained image processing Transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12299–12310
5. Chen L, Chu X, Zhang X et al (2022) Simple baselines for image restoration. In: Computer vision-ECCV 2022: 17th European conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII. Springer, pp 17–33
6. Chen X, Wang X, Zhou J et al (2023) Activating more pixels in image super-resolution transformer. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 22367–22377
7. Chu X, Chen L, Yu W (2022) NAFSSR: Stereo image super-resolution using NAFNet. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 1239–1248
8. Conde MV, Choi UJ, Burchi M et al (2022) Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In: European conference on computer vision. Springer, pp 669–687
9. Dai Q, Li J, Yi Q et al (2021) Feedback network for mutually boosted stereo image super-resolution and disparity estimation. In: Proceedings of the 29th ACM international conference on multimedia, pp 1985–1993
10. Dai T, Cai J, Zhang Y et al (2019) Second-order attention network for single image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 11065–11074
11. De S, Smith S (2020) Batch normalization biases residual blocks towards the identity function in deep networks. Adv Neural Inf Process Syst 33:19964–19975
12. Dong C, Loy CC, He K et al (2015) Image super-resolution using deep convolutional networks. IEEE Trans Pattern Anal Mach Intell 38(2):295–307
13. Dosovitskiy A, Beyer L, Kolesnikov A et al (2021) An image is worth 16x16 words: transformers for image recognition at scale. ICLR
14. Geiger A, Lenz P, Urtasun R (2012) Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition, IEEE, pp 3354–3361
15. Hassani A, Walton S, Li J et al (2022) Neighborhood attention transformer. arXiv:2204.07143
16. Hendrycks D, Gimpel K (2016) Gaussian error linear units (GELUs). arXiv:1606.08415
17. Hu J, Shen L, Sun G (2018) Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7132–7141
18. Hua Y, Kohli P, Uplavikar P et al (2020) Holopix50k: a large-scale in-the-wild stereo image dataset. In: CVPR workshop on computer vision for augmented and virtual reality, Seattle, WA
19. Huang G, Sun Y, Liu Z et al (2016) Deep networks with stochastic depth. In: European conference on computer vision. Springer, pp 646–661
20. Loshchilov I, Hutter F (2017) SGDR: stochastic gradient descent with warm restarts. In: International conference on learning representations. https://openreview.net/forum?id=Skq89Scxx
21. Jeon DS, Baek SH, Choi I et al (2018) Enhancing the spatial resolution of stereo images using a parallax prior. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)
22. Jiang K, Wang Z, Yi P et al (2020) Hierarchical dense recursive network for image super-resolution. Pattern Recognit 107:107475. https://doi.org/10.1016/j.patcog.2020.107475. https://www.sciencedirect.com/science/article/pii/S0031320320302788
23. Johnson J, Alahi A, Fei-Fei L (2016) Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision, Springer, pp 694–711
24. Kim J, Lee JK, Lee KM (2016) Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1646–1654
25. Ledig C, Theis L, Huszár F et al (2017) Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4681–4690
26. Lei J, Zhang Z, Fan X et al (2021) Deep stereoscopic image super-resolution via interaction module. IEEE Trans Circ Syst Video Technol 31(8):3051–3061. https://doi.org/10.1109/TCSVT.2020.3037068
27. Li B, Lin CW, Shi B et al (2018) Depth-aware stereo video retargeting. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6517–6525
28. Liang J, Cao J, Sun G et al (2021) SwinIR: image restoration using swin Transformer. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV) workshops, pp 1833–1844
29. Lim B, Son S, Kim H et al (2017) Enhanced deep residual networks for single image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 136–144

30. Lin J, Yin L, Wang Y (2023) Steformer: efficient stereo image super-resolution with transformer. IEEE Trans Multimed 25:8396–8407. https://doi.org/10.1109/TMM.2023.3236845
31. Liu A, Li S, Chang Y et al (2024) Coarse-to-fine cross-view interaction based accurate stereo image super-resolution network. IEEE Trans Multimed 1–13. https://doi.org/10.1109/TMM.2024.3364492
32. Liu Z, Lin Y, Cao Y et al (2021) Swin Transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 10012–10022
33. Liu Z, Li Z, Wu X et al (2022) DSRGAN: detail prior-assisted perceptual single image super-resolution via generative adversarial networks. IEEE Trans Circ Syst Video Technol 32(11):7418–7431
34. Loshchilov I, Hutter F (2018) Decoupled weight decay regularization. In: International conference on learning representations
35. Mei Y, Fan Y, Zhou Y et al (2020) Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 5690–5699
36. Mei Y, Fan Y, Zhou Y (2021) Image super-resolution with non-local sparse attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 3517–3526
37. Menze M, Geiger A (2015) Object scene flow for autonomous vehicles. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3061–3070
38. Scharstein D, Hirschmüller H, Kitajima Y et al (2014) High-resolution stereo datasets with subpixel-accurate ground truth. In: German conference on pattern recognition. Springer, pp 31–42
39. Schops T, Schonberger JL, Galliani S et al (2017) A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3260–3269
40. Shi W, Caballero J, Huszár F et al (2016) 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, pp 1874–1883
41. Timofte R, De Smet V, Van Gool L (2013) Anchored neighborhood regression for fast example-based super-resolution. In: Proceedings of the IEEE international conference on computer vision, pp 1920–1927
42. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need. In: Advances in neural information processing systems, pp 5998–6008
43. Wang L, Wang Y, Liang Z et al (2019) Learning parallax attention for stereo image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 12250–12259
44. Wang X, Girshick R, Gupta A et al (2018) Non-local neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7794–7803
45. Wang X, Yu K, Wu S et al (2018) ESRGAN: enhanced super-resolution generative adversarial networks. In: Proceedings of the European conference on computer vision (ECCV) workshops, pp 0–0
46. Wang Y, Wang L, Wang H et al (2018) Resolution-aware network for image super-resolution. IEEE Trans Circ Syst Video Technol 29(5):1259–1269
47. Wang Y, Wang L, Yang J et al (2019) Flickr1024: a large-scale dataset for stereo image super-resolution. In: International conference on computer vision workshops, pp 3852–3857
48. Wang Y, Ying X, Wang L et al (2021) Symmetric parallax attention for stereo image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 766–775
49. Wang Z, Bovik AC, Sheikh HR et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
50. Wang Z, Cun X, Bao J et al (2022) Uformer: a general u-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 17683–17693
51. Wu H, Zou Z, Gui J et al (2020) Multi-grained attention networks for single image super-resolution. IEEE Trans Circ Syst Video Technol 31(2):512–522
52. Yan B, Ma C, Bare B et al (2020) Disparity-aware domain adaptation in stereo image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)
53. Yang J, Wright J, Huang TS et al (2010) Image super-resolution via sparse representation. IEEE Trans Image Process 19(11):2861–2873
54. Zamir SW, Arora A, Khan S et al (2022) Restormer: efficient transformer for high-resolution image restoration. In: CVPR
55. Zhang H, Cisse M, Dauphin YN et al (2018a) Mixup: beyond empirical risk minimization. In: International conference on learning representations. https://openreview.net/forum?id=r1Ddp1-Rb
56. Zhang J, Long C, Wang Y et al (2021) A two-stage attentive network for single image super-resolution. IEEE Trans Circ Syst Video Technol 32(3):1020–1033

57. Zhang W, Liu Y, Dong C et al (2019) RankSRGAN: generative adversarial networks with ranker for image super-resolution. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 3096–3105
58. Zhang Y, Li K, Li K et al (2018) Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European conference on computer vision (ECCV), pp 286–301
59. Zhang Y, Tian Y, Kong Y et al (2018) Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2472–2481
60. Zhang Z, Peng B, Lei J et al (2022) Recurrent interaction network for stereoscopic image super-resolution. IEEE Trans Circ Syst Video Technol
61. Zhu X, Guo K, Fang H et al (2021) Cross view capture for stereo image super-resolution. IEEE Trans Multimed. https://doi.org/10.1109/TMM.2021.3092571
62. Zhu Z, Xu M, Bai S et al (2019) Asymmetric non-local neural networks for semantic segmentation. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 593–602
63. Qian N (1999) On the momentum term in gradient descent learning algorithms. Neural networks 12(1):145–151
64. Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of machine learning research 12(7)
65. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980