scientific reports

Check for updates

OPEN

Stereoscopic video deblurring transformer

Hassan Imani[1], Md Baharul Islam[1,2]✉, Masum Shah Junayed[1,3] & Md Atiqur Rahman Ahad[4]✉

Stereoscopic cameras, such as those in mobile phones and various recent intelligent systems, are becoming increasingly common. Multiple variables can impact the stereo video quality, e.g., blur distortion due to camera/object movement. Monocular image/video deblurring is a mature research field, while there is limited research on stereoscopic content deblurring. This paper introduces a new Transformer-based stereo video deblurring framework with two crucial new parts: a self-attention layer and a feed-forward layer that realizes and aligns the correlation among various video frames. The traditional fully connected (FC) self-attention layer fails to utilize data locality effectively, as it depends on linear layers for calculating attention maps The Vision Transformer, on the other hand, also has this limitation, as it takes image patches as inputs to model global spatial information. 3D convolutional neural networks (3D CNNs) process successive frames to correct motion blur in the stereo video. Besides, our method uses other stereo-viewpoint information to assist deblurring. The parallax attention module (PAM) is significantly improved to combine the stereo and cross-view information for more deblurring. An extensive ablation study validates that our method efficiently deblurs the stereo videos based on the experiments on two publicly available stereo video datasets. Experimental results of our approach demonstrate state-of-the-art performance compared to the image and video deblurring techniques by a large margin.

Video deblurring is the process of restoring acute frames out of a blurry video. Deblurring is a crucial foundation for many computer vision tasks, and has therefore attracted significant research interest. Camera shake and object movement are common blur artifacts in dynamic video scenes[1,2]. In video processing, movement is critical, which causes most of the blur in a video, known as motion blur. Most approaches in this field first compute the motion between successive frames before applying frame transformations[3,4]. Consequently, the efficiency of the motion estimation profoundly influences the whole method's functionality. Precise motion prediction, on the other hand, is complex and time-consuming[5]. Furthermore, most motion estimation algorithms address an optimization issue, slowing motion estimation. Some approaches use generative networks for video deblurring. For instance, Fanous et al.[6] employed a generative adversarial network (GAN) for frame deblurring.

Limited research is reported in the literature for stereo video deblurring. In a recursive architecture, Pan et al.[7] used stereo view information that a coarser depth or scene flow is used to calculate blur kernels. Some other studies employed stereo disparity and video motion. They estimated the disparity using data from the stereoviews and suggested a region tree technique for calculating the point spread functions (PSFs). Sellent et al.[8] mention scene flow and stereo video deblurring as typical issues. Local homographs were employed to produce blur kernels using scene flow calculations, and scene flow and deblurring were addressed separately using pre-estimated scene flow.

Stereo video deblurring requires to preserve both disparity and temporal coherence. This makes it different from applying regular deblurring methods used for single images or standard videos. As a result, the motion information within successive frames potentially plays a considerable part in deblurring the frames next to them. Therefore, stereo video deblurring work can be divided into two significant components: (a) modeling symmetry cues across two viewpoints and (b) simulating sequences among subsequent frames. The intrinsic relation across pairs of stereo frames is exploited for modeling symmetry. Two considerations lead to our desire to propose a novel methodology for stereo video deblurring. Firstly, utilizing the motion information across succeeding frames and combining the information from adjacent frames of one perspective can aid in detecting distortions in pixels of the center frame. In fact, due to the slight movement between the few subsequent frames, surrounding frames can assist in deblurring the desired frame when deblurring a single video frame. Secondly,

[1]Faculty of Engineering and Natural Sciences, Bahcesehir University, 34353 Istanbul, Turkey. [2]Department of Computing and Software Engineering, Florida Gulf Coast University, Fort Myers, FL 33965, USA. [3]Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA. [4]Department of Computer Science and Digital Technologies, University of East London, London, UK. ✉email: mislam@fgcu.edu; mahad@uel.ac.uk

stereo vision provides two views simultaneously. Using the depth map, the equivalent pixels in one viewpoint can aid in the removal of blur in the comparable stereo view.

The transformer[1] is well-known because of its capabilities in parallelization and outstanding modeling ability of the interconnections between the input sequences. It can potentially handle stereo video enhancement as a sequence modeling task[9]. Transformer-based approaches, such as Vision Transformers (ViT)[10], break a video sequence into tiny areas and derive global connections among the token embeddings that reflect the areas. At the same time, spatial information is not granted considerable weight[2]. Such frameworks can only be used in a way that allows for stereo video deblurring, relying on local and texture information. Moreover, the ViT is not designed to resolve temporal dependencies and consistency, which are critical in the stereo-deblurring challenge.

To deal with motion blur, this study provides a novel Transformer-based stereo video deblurring approach that leverages nearby frames and information from the other corresponding stereo frame. Our Transformer-based stereo video deblurring approach leverages nearby frames and information from corresponding stereo frames to handle temporal information. We design an optical flow-based feed-forward layer to discover correlations across different video frames and align the features. Our approach employs a combination of spatial and temporal attention mechanisms to capture both local and global dependencies across frames. Specifically, we utilize a self-attention mechanism within each frame to model relationships between pixels, addressing spatial attention. Additionally, we introduce an optical flow-based feed-forward layer as a temporal attention mechanism to model relationships between consecutive frames, aiding the model in understanding the dynamics of the video sequence. By combining these two attention mechanisms, our architecture effectively captures both spatial and temporal dependencies in videos. We first estimate the motion information between consecutive frames using PWC-Net[11] model. Then, after applying a 3D convolution, we perform a Transformer network to both stereo views. Then, the extracted features are fed to a CNN-based unit, and the features from the stereo frames are fused using a modified Parallax Attention Mechanism (mPAM) module. Lastly, a reconstruction layer creates the deblurred targeted frames. Due to the usage of both inter-view and intra-view frames, the temporal information of the video are handled in our method. The primary contributions to this paper are given below:

- We propose a new transformer model for deblurring stereoscopic videos. To deblur a target frame, the presented model incorporates the cross-view information and the information from nearby frames.
- In the model, we present a new feed-forward layer that spatially aligns features by calculating the relationships among all neighboring frames.
- We significantly improved the PAM module, namely mPAM, for combining features from stereo views to merge the stereo video features.
- Several image- and video-based deblurring methods are reimplemented to have a fair comparison with the proposed method based on two benchmark datasets. Experimental results and ablation studies show the superiority of our method compared to the previous art.

In Section "Related works", we briefly illustrate essential methods related to 2D and 3D images and video deblurring. We describe the proposed model and its different parts in Section "Proposed method". Section "Datasets and experiments" discusses the experimental setup, implementation, and datasets. The efficiency of our method is evaluated in Section "Results and discussions". Finally, we conclude the paper with some future work guidelines.

Related works

This section briefly discusses the relevant single, stereo image, and video deblurring methods.

2D image deblurring

Certain classic methods for removing the blur from a single image are proposed and available in the literature. Some examples include the L0 regularized prior[12], the dark channel prior[13], and the discriminative prior[14]. These methods have several limitations in representing spatial blur in dynamic settings. These methods often struggle to represent complex, spatially-varying blur in dynamic scenes with motion. However, several methods, including[15–17], used the depth map to simulate the blur distortion that is not homogeneous. Because of the time-consuming optimization process, such methods are expensive.

Traditional deblurring methods are computationally expensive[18]. For dealing with commonly occurring blur resulting from the relative movement of the object-camera, Nah et al.[19] developed a no-reference solution. This method is a CNN-based multi-scale system that attempts to recover frames with more details. The approach suggested in[20] involves gradually recovering the image at various qualities from providing a strategy that is less complicated than earlier techniques and performs better. A multi-scale structure has been included in the suggested paradigm. Zhang et al.[21] presented a strategy for dealing with the spatially variable blur, which occurs as the camera moves. Three CNNs and one RNN were employed. Liang et al.[22] approached the deblurring problem from another perspective. They proposed a new model for deblurring raw images. They also used a new raw image deblurring dataset and trained their model on that dataset. In another study, Honorvar et al.[23] proposed a new model of PSF of motion blur to analyse the motion invariant in frequency and moment domains.

If the blur is not uniformly distributed, for example[24,25], employed CNNs to predict the blurry regions. In[26], the authors developed a new approach for detecting motion blur caused by camera and object movements. They designed a new multi-scale CNN-based framework with certain skip connections to manage data generation. Recently, an Edge-Aware Scale-Recurrent Network (EASRN) was presented by Chang et al.[27] to deal with the motion blur in the presence of the outliers that deblurred the frames at different scales. This method also trained a deep model to restore the high-quality edges. Li et al.[28] developed a CNN-based model for image deblurring based on depth information estimation. Then, they use a feature transform model to extract depth features and

combine them with spatial features. It demonstrated that depth information could be effectively utilized for image deblurring. Very recently, Restformer[29] is proposed which is a Transformer-based model. They design multi-head attention and feed-forward blocks to capture long-range pixel interactions. In another recent study, Kong et al.[30] introduced a frequency domain-based Transformer for deblurring images. Instead of matrix multiplication, they calculate the scaled dot-product attention using their proposed product method.

2D video deblurring

Several recent works have addressed 2D video deblurring[31–36]. Delbracio et al.[31] used the Fourier transform to fuse the data from neighboring frames in a video to remove motion blur. The neighboring frames are registered for each frame, and then the registered frames are combined using the Fourier transform. CNN's are one of the most successful methods developed for video deblurring. For example, an encoder-decoder-based model is applied to the batch of neighboring frames for deblurring in[32]. The method in[33] proposed a Spatio-temporal 3D CNN model to deblur videos. Zhang et al.[36] modeled the temporal dependencies using a non-local layer that calculated the similarities and differences between frames with a recursive block.

Pan et al.[34] proposed an optical flow-based model in another study. This method learns CNN to calculate the optical flow and reconstructs the deblurred frames afterward. Son et al.[35] are also based on using neighboring frames. They proposed a novel motion estimation method that is invariant to blur. Instead of warping frames for compensating motion, they used a pixel volume to to use the most sensitive pixels of the blurred video. Recently, Wang et al.[37] presented a CNN-based model, providing spatial-temporal and frame channel attention modules and a reconstruction block to re-create the high-resolution frames. Video deblurring and optical flow (VDFlow)[38] estimated optical flow and deblurring at the same time. This model has two parts: encoder-decoder for deblurring and optical flow network (FlowNet)[39] for optical flow estimation. In another study, Chen et al.[40] formulated deblurring as a residual learning problem. They trained a U-net model to deblur the frames and then iteratively generated frames to create a high frame-rate video.

Stereo image and video deblurring

Some studies have employed disparity and motion (for video) to deblur stereo content. The depth information and point spread functions were calculated in[41]. They estimated the depth of information and then suggested a region tree approach for computing the point spread functions[8] used scene flow estimates to generate blur kernels and a grading approach to the borders of moving objects. In contrast, Pan et al.[7] combined scene flow estimation with deblurring and discovered that motion and blur distortions could interact. Network with depth awareness and view aggregation (DAVANet)[42] was proposed for stereo image deblurring. It includes three major sections: an encoder-decoder backbone, a disparity prediction model, and an integration framework that combines the two networks to generate deblurred frames. They also presented the Stereo Blur dataset. Recently, UNet-Deblur[43] introduced a CNN-based stereo video deblurring approach that considered the stereo frames in succession. They fed the target and successive neighboring frames to the 3D CNN model to adjust for motion in stereoscopic video, which can aid with more deblurring. After compensating for motion across subsequent frames, the left and right frames are subjected to a 3D CNN to extract their features. They redesigned 3D U-Nets to use them as feature extractors. The PAM[44] module is adjusted to fuse cross-view information and construct the output deblurred frames to combine the left and right information. Besides, despite having deeper architecture compared to the other stereo image-based methods such as DAVANet[42], their method has poor efficiency. Motivated by this, we develop a new architecture to better utilize the neighboring and stereo information to deblur the stereo video frames efficiently.

Proposed method

Figure 1 shows the design architecture of our stereo video deblurring approach. We estimate the motion between succeeding center frames using the pyramid, warping, and cost volume network (PWC-Net)[11]. After warping the neighboring frames to the center frames, we apply them into a 3D convolutional block, which extracts even more localized characteristics. A Transformer network then learns the features from the middle and motion-compensated frames. We use four convolutional residual blocks (CRB) to extract more deep features. The CRB provides features with broad receptive fields and intense sampling rates, which help to estimate stereoscopic matching. Then, we combine the cross-view features with modifying the PAM[44]. Finally, a batch of 2D convolutional blocks reconstructs the target frames and further adds the middle frames. We first discuss PWC-Net Architecture, and then we discuss the proposed Transformer Architecture.

PWC-Net and transformer architecture

PWC-Net Architecture

We utilize PWC-Net, which is built upon fundamental principles: pyramidal processing, warping, and leveraging a cost volume. Implemented within a trainable feature pyramid, PWC-Net utilizes the existing optical flow estimation to deform the CNN features of the subsequent image. It then combines these deformed features with those from the initial image to create a cost volume. This volume is then analyzed by a CNN to estimate the optical flow. Optical flow approximation is fundamental in vision tasks with several use cases[45]. The energy reduction strategy proposed by Horn and Schunck[46] is used by state-of-the-art approaches. Nevertheless, optimizing a complicated energy function is typically costly for real-world use cases. Figure 2 summarizes the major parts of PWC-Net. First, we calculate the feature pyramids to extract features at different scales. Let ($\mathbf{I}_t^l$ and $\mathbf{I}_t^r$) and ($\mathbf{I}_{t-1}^l$ and $\mathbf{I}_{t-1}^r$) represent the two stereo consecutive frames. Pyramid extraction includes six levels, with 16, 32, 64, 96, 128, and 196 number of features[11]. The calculated pyramids are as follows: $\mathbf{P}_t^l$, l=0,...,5. Then, another layer performs the

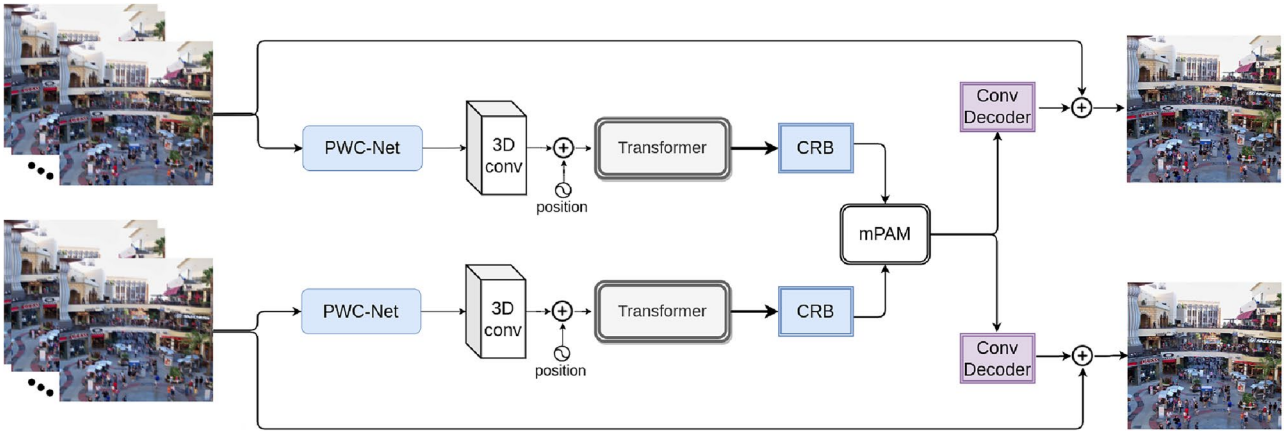

Figure 1. The proposed stereo video deblurring model. Firstly, PWC-Net estimates the motion between the neighboring frames. Then, we apply a 3D CNN layer to the motion-compensated frames, and the proposed Transformer model accepts the resulting features as input. Next, another CNN layer (CRB) extracts deep features. The mPAM then fuses the stereo input features. A convolutional decoder constructs the deblurred frames from the left and right features. Finally, we form the output by adding the blurry middle target frames with the reconstructed left and right frames.

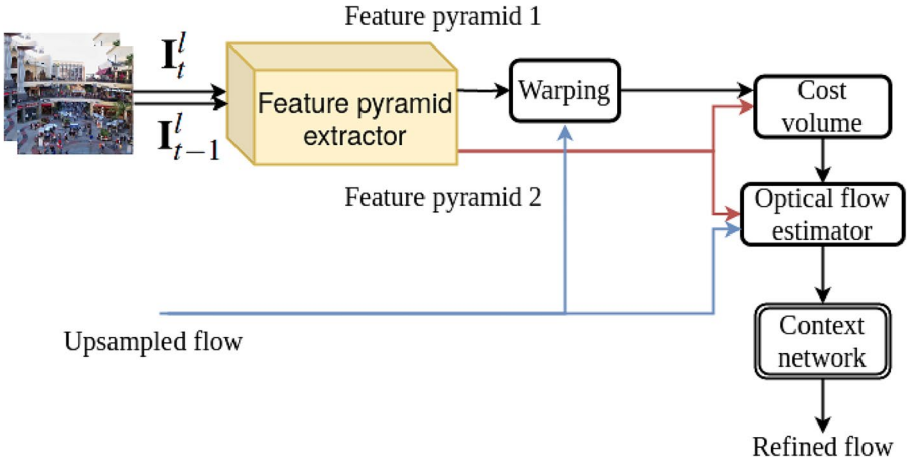

Figure 2. Feature pyramid in PWC-Net[11]. The arrows represent the flow estimation direction, while the pyramids are built in reverse directions. PWC-Net uses the upsampled flow to warp features in the neighboring frame, calculates a cost volume, and processes it with neural networks.

warping process. For stereo frames, the features of $\mathbf{I}_t^l$ and $\mathbf{I}_t^r$ are warped using the features of $\mathbf{I}_{t-1}^l$ and $\mathbf{I}_{t-1}^r$, and the up-sampled flow of the upper pyramid level from the l+1th level for each view:

$$P_t^l(i) = P_{t-1}^l(i + up(t^{l+1})(i)) \tag{1}$$

In this equation, i and up are the pixel index and the upsample operators, respectively. Here, the bilinear interpolation calculates the warps.

Figure 3 depicts the Transformer's high-level architecture. Firstly, we apply a 3D CNN to the stereo batches to transfer the input frames ($\mathbf{I}_3^{lcomp}$ and $\mathbf{I}_3^{rcomp}$) from 3 to 64 output channel ($\mathbf{I}_{64}^{lcomp}$ and $\mathbf{I}_{64}^{rcomp}$). Next, we calculate the initial features using residual modules ($\mathbf{I}_{64}^{lRes}$ and $\mathbf{I}_{64}^{rRes}$). The added and normalized blocks connect attention and flow with residual layers. As seen in Fig. 3, we repeat these layers L times and apply another residual block. We discuss the transformer's sub-blocks in the following sub-sections.

Self-attention layer

Figure 4 depicts the architecture of this layer. We start with creating the Query (Q), Key (K), and Value (V) tensors. With applying a 3D CNNs to ($\mathbf{I}_{64}^{lRes}$ and $\mathbf{I}_{64}^{rRes}$), we generate Q (Q_{64}^l and Q_{64}^r) and K tensors (K_{64}^l and K_{64}^r) to extract their feature maps. 64 filters with size of 3×3×3 and padding of 1 perform to 3 CNNs. Therefore, Q, K, and V for the left channel are as follows:

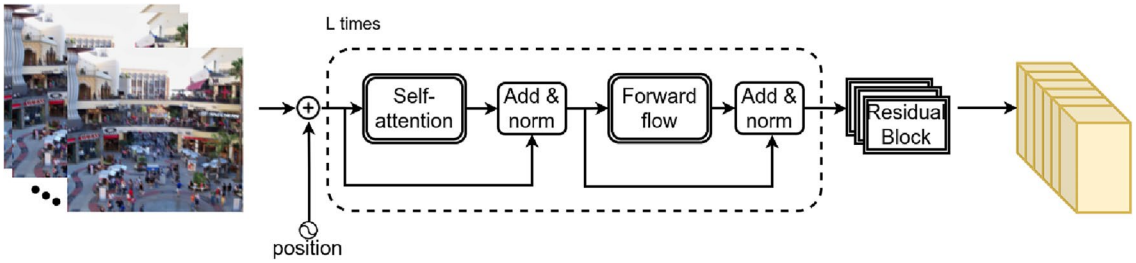

Figure 3. The Transformer's high-level design structure. To extract information from the frames, we use convolutional layers. The self-attention and feed-forward optical flows are applied after position encoding, utilizing the add and normalization blocks. Finally, residual modules create the desired outputs.

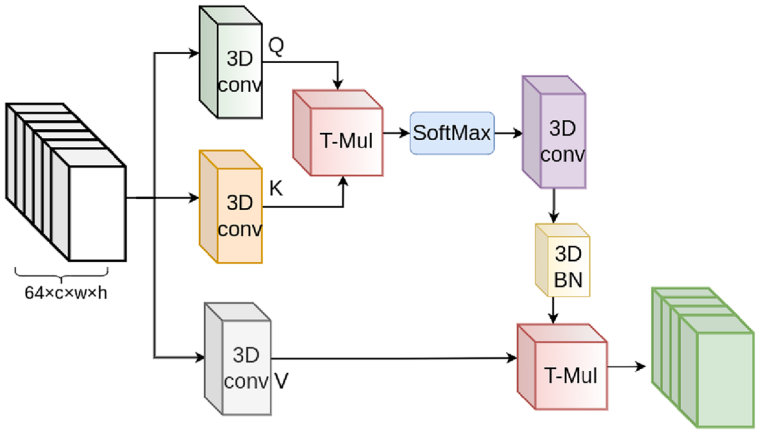

Figure 4. The self-attention module's architecture. The input features from a 3D CNN module build the tensors Q, K, and V, and after tensor multiplications, we create the output.

$$\begin{aligned} Q &= 3DCNN(K_1, \mathbf{I}_{64}^{l_{Res}}) \\ K &= 3DCNN(K_2, \mathbf{I}_{64}^{l_{Res}}) \\ V &= 3DCNN(K_3, \mathbf{I}_{64}^{l_{Res}}) \end{aligned} \tag{2}$$

where $K_{1,2,3}$ are CNN kernels. Next, we calculate the similarity tensor using the tensor product (TP) for the left video:

$$QK_l = SM(TP(Q^T, K)) \tag{3}$$

where SM is the softmax operation. We apply the output features into a 3D CNN including 64 filters and 3×3×3 kernel size. Next, we multiply the results by V and combine them with the input features to obtain the attention layer's output features for the left video:

$$Attn_l = \mathbf{I}_{64}^{l_{Res}} + TP(QK_l, V_l) \tag{4}$$

The calculations for the right features are identical to the left one.

Position encoding

The permutation is unchanging in the original Transformer architecture[47], but in deblurring task, the position is crucial. In this paper, we use the positional encoding in[48]. For left and right Transformers, we utilize $d/3$ sine and cosine with distinct frequencies for each spatial coordinate:

$$PE_l(pos_l, i) = \begin{cases} sin(pos_l.w_k) & \text{for} \quad i = 2k, \\ cos(pos_l.w_k) & \text{for} \quad i = 2k+1; \end{cases} \tag{5}$$

where pos_l is the position in the dimension for the left Transformer, and $w_k = 1/10000^{2k/(d/3)}$[48].

Feed-Forward (FF) Layer

The fully connected FF does not utilize the interdependence across tokens of neighboring frames. We propose an optical flow-based approach to align the input features in the spatial dimension, considering the relations between successive frames. Figure 5 describes the proposed architecture. We apply the feature maps from $Attn_l$

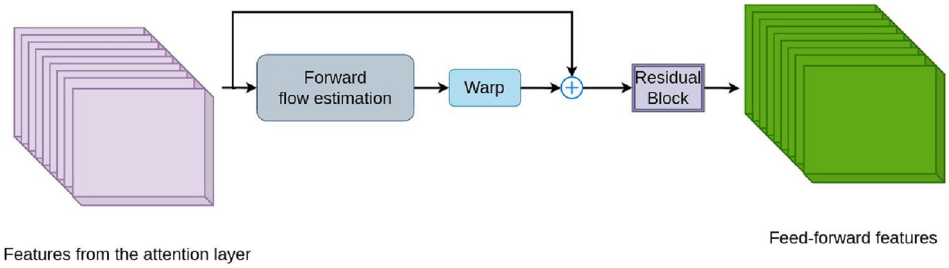

Figure 5. Architecture of the optical flow-based feed-forward layer: Firstly, the features coming from the self-attention layer estimate the forward optical flows. Then, after the warping operation, residual and convolutional layers create the output features.

and $Attn_r$ to this block. We use spatial pyramid network (SpyNet)[49] to estimate the motions across frames n and m as $flow_l$ and $flow_r$:

$$\begin{aligned} flow_l(m,n) &= \begin{cases} [0]_{W\times H} & \text{for} \quad m=n, \\ spy(\mathbf{I}_n^l, \mathbf{I}_m^l) & \text{for} \quad m\neq n; \end{cases} \\ flow_r(m,n) &= \begin{cases} [0]_{W\times H} & \text{for} \quad m{=}n, \\ spy(\mathbf{I}_n^r, \mathbf{I}_m^r) & \text{for} \quad m\neq n; \end{cases} \end{aligned} \tag{6}$$

where spy is the SpyNet[49], and LR Next, we warp the features in the forward direction:

$$\begin{aligned} FF_l &= warp(Attn_l, flow_l) \\ FF_r &= warp(Attn_r, flow_r) \end{aligned} \tag{7}$$

Next, we combine the FF_l and FF_r with $Attn_l$ and $Attn_r$. To build the connection between succeeding frames, we suggest using a CNN-based forward layer. To construct the resulting features of this module, we particularly employ residual blocks with a 3D CNN at the end. The following is how we define a fully connected feed-forward layer:

$$\begin{aligned} FF_l^o(Attn_l) &= conv(LN(Attn_l + Res([Attn_l, FF_l]))) \\ FF_r^o(Attn_r) &= conv(LN(Attn_r + Res([Attn_r, FF_r]))) \end{aligned} \tag{8}$$

Modified PAM (mPAM)

Stereo video frame pairs offer an opportunity to enhance the effectiveness of image and video deblurring by providing supplementary information from a second perspective. Nonetheless, integrating this data presents challenges due to the considerable variations in disparities between stereo images. To address this, we propose a parallax-attention mechanism (PAM) featuring a global receptive field along the epipolar line. This mechanism aims to manage diverse stereo video frames with substantial differences in disparity effectively. Parallax Attention Mechanism (PAM)[44] merges the features of stereo images. We improve the PAM design to account for the input 3D features representing video sequences over time. The input features to the mPAM module are 3 dimensional (from left or right videos). Therefore, 3d residual features at first, then apply 2D convolutions. As shown in Fig. 6, the left and right features are fed to the 3D residual blocks (Res). 2D convolutions (2D conv)

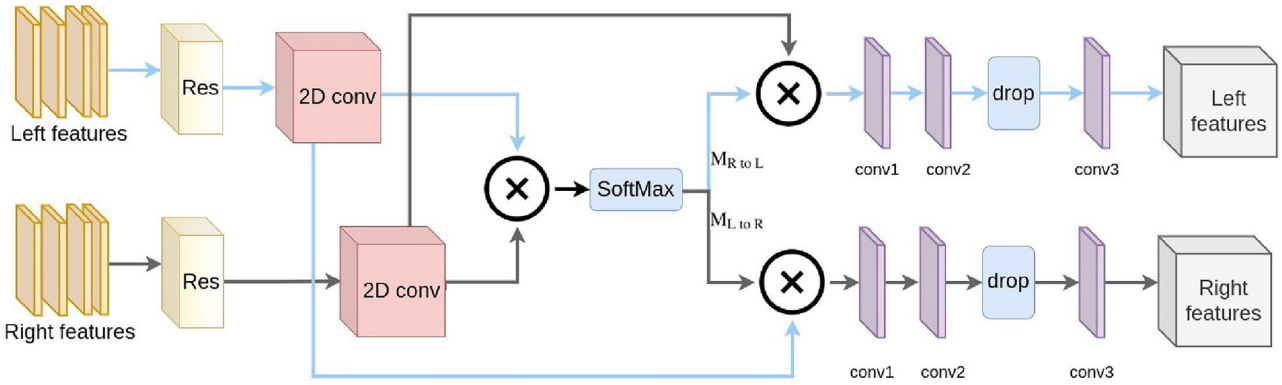

Figure 6. The mPAM flow diagram: Firstly, the stereo input features are input to the residual layer (Res). After applying a 2D CNN, we fuse the cross-view information and create the output.

are applied next to make the input suitable for 3D features. Tensor multiplication is then performed to the left and right features. SoftMax block then creates the attention maps: $M_{R\ to\ L}$ (from right to left) and $M_{L\ to\ R}$ (from left to right). Next, for all disparities, we combine the summation of features with the former right features. We removed valid mask generation from PAM structure in[44], because the authors use an occlusion detection method to generate valid masks. Since this operation adds to the computations, we removed it from the main algorithm. To generate deeper features suitable for deblurring, we utilize 3 CNN layers. There are 128 filters in the initial 2D convolution *conv1*. For this convolution, we employed a 5×5 kernel size. Just by changing the kernel size to 3×3, the *conv2* is similar to the *conv1*. Then, at a rate of 0.5, we apply a dropout layer *drop*. The third layer *conv3* is with 64 filters and a 3×3 kernel.

Loss functions

Five loss functions are defined in this section which we use for model training. The mean absolute error (MAE) is the first loss, which determines the differences among the original and deblurred frames. The average MAE of the stereo viewpoints is as follows:

$$mae_{loss} = (mae_l + mae_r)/2 \tag{9}$$

In addition, we exploit photometric (p_{loss}) and cycle (c_{loss}) losses[44]. To consider the smoothness in correspondence space, we use smoothness loss as follows:

$$\begin{aligned} s_{loss} = \sum_{A} \sum_{i,j,k} (||A(i,j,k) - A(i+1,j,k)||_1 \\ + ||A(i,j,k) - A(i,j+1,k+1)||_1) \end{aligned} \tag{10}$$

where, A is the cross-view attention maps. Finally, stereo consistency loss $sConsist_{loss}$ considers the stereo consistency between deblurred stereo frames. For stereo consistency, we calculate the end-point error (EPE) using Euclidean distance among the two disparities of the original and deblurred video frames. The resulting loss is as the union of defined five losses:

$$\begin{aligned} loss = mae_{loss} + \gamma (p_{loss} + c_{loss} + s_{loss} \\ + sConsist_{loss}) \end{aligned} \tag{11}$$

where, γ is a constant which is set as 0.05.

Datasets and experiments

To train the proposed deblurring model, we utilize the only publicly available dataset of the Stereo Blur dataset[42]. For model evaluation, we use the test set of Stereo Blur and LFOVIAS3DPh2[50] datasets that are discussed in the following subsections.

Datasets and evaluation criteria

Stereo blur [42] dataset

This dataset contains videos of objects and people with minor disparities. The outdoor videos include humans, cars, boats, and outdoor scenarios. Furthermore, the dataset contains videos captured in various situations, such as lighting and weather variations. The authors expanded the dataset to include a variety of motion settings utilizing three distinct imaging styles: handheld, stationary, and onboard shots. The ZED stereo camera[51] is being used to create this dataset, with an FPS of 60. The stereoscopic video has identical arrangements on both stereo sides. It includes masks for eliminating faulty samples in the disparity and distorted frame segments, generated using the bidirectional consistency check[52]. In this dataset, there are 135 stereo videos.

LFOVIAS3DPh2 [50] Dataset

It is used for stereoscopic video quality assessment[53–55] and contains 12 pure and 288 distorted videos. These videos were recorded with a Panasonic camera, and their resolution is 1920 × 1080. High-quality videos are labeled with a high value and vice versa (ranging from 5 for the highest quality and 0 for the lowest grade). All the videos have an exact duration of 10 seconds. Since the LFOVIAS3DPh2 dataset contains blurry and original videos, we use this dataset's blurry videos to evaluate our stereo video deblurring method. To make blurry videos, the authors in[50] employed ffmpeg's *box blur* function. They created 72 blurry stereo videos by applying 3 blur levels to the 12 reference stereo videos.

Evaluation metrics

We compare our model's performance to deep learning-based and classical approaches in the two popular Structural SIMilarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) metrics.

Experimental setup

To train the proposed model, we firstly center crop the left and right frames with 256 pixels and construct a dataset with a size of 256 × 256. Our computing system's configurations are NVIDIA RTX 3090 GPU, 24GB of GPU RAM, and i9-10850K CPU 3.60 GHz. We utilize the Adam optimizer[56] with β_1=0.9 and β_2=0.99. We employ a batch size of 10 with the learning rate of 0.001, and we trained the model for 528k iterations.

Results and discussions

To our best knowledge, only UNet-Deblur[43] as a video-based stereo deblurring method reported results on the Stereo Blur dataset. As a result, we do comparisons with this method, stereo image-based approaches, and some video and image deblurring methods. In Zhou et al.[42], the models of[19–21,57] are trained on the Stereo Blur dataset. Tables 1 and 2 demonstrate the outcomes of the analysis of image- and video-based deblurring approaches for Stereo Blur and LFOVIAS3DPh2[50] datasets, respectively.

Quantitative results

We compare the proposed method's effectiveness with the available 2D and 3D image and video-based methods in Table 1, notably the only available stereo video deblurring method[43]. The results demonstrate that our model improved by 3.50 dB in PSNR and 0.0521 dB in SSIM, which significantly improved. Furthermore, stereo video deblurring approaches of Sellent et al.[8] and Pan et al.[7] are not open-source, and their results on the Stereo Blur dataset have not been published. They conducted their research using videos that they created for their experiments. Sellent et al.[8] created stereo images for their experiments, which is not possible to use in our experiments since our method requires some successive frames. Our algorithm requires at least 5 successive frames. In addition, it contains a few images, which means it cannot train our deep learning-based model. Since the training code for[7] is not available, we could not compare our results with it. To facilitate comparison, we re-implemented two 2D video deblurring approaches of Son et al.[35] and Pan et al.[34]. Pan et al.[34] efficiently use domain knowledge of video deblurring. Still, our method outperforms this method thanks to using the mPAM module. Compared to Son et al.[35] model, we improve 0.83 and 0.27 dB in PSNR on Stereo Blur and LFOVIAS3DPh2 datasets, respectively. DAVANet[42] is a stereo image deblurring method that performs better than the other image-based methods by a large margin. We also compare PAM[44] with the proposed mPAM inside our whole model. Table 5

Methods	PSNR	SSIM	Time (s)	Params (M)
Image-based Methods				
Whyte[58]	24.48	0.8410	700	–
Sun[24]	26.13	0.8830	1200	7.26
Gong[25]	26.51	0.8902	1500	10.29
Nah[19]	30.35	0.9294	4.78	11.71
Kupyn[57]	27.81	0.8895	0.22	11.38
Zhang[21]	30.46	0.9367	1.40	9.22
Tao[20]	31.65	0.9479	2.52	8.06
DAVANet[42]	33.19	0.9586	0.31/pair	8.68
2D Video-based Methods				
Pan et al.[34]	33.78	0.9572	0.42/pair	32.4
Son et al.[35]	33.22	0.9328	0.25/pair	21.02
Stereo Video-based Methods				
UNet-Deblur[43]	30.56	0.9221	0.57/pair	19.9
Ours	**34.06**	**0.9742**	0.81/pair	38.4

Table 1. Comparison of our proposed method with image- and video-based deblurring methods in terms of PSNR, SSIM, and time-complexity on the Stereo Blur[42] dataset. The best results are in bold. The "–" is used for unavailable information.

Methods	PSNR	SSIM
Image-based methods		
Kupyn[57]	27.12	0.8770
DAVANet[42]	32.1073	0.9394
2D video-based methods		
Pan et al.[34]	32.0331	0.9387
Son et al.[35]	31.9880	0.9355
Stereo video-based methods		
UNet-Deblur[43]	28.7216	0.9018
Ours	**32.2601**	**0.9410**

Table 2. Comparison of our proposed method with image- and video-based deblurring methods in terms of PSNR and SSIM on the LFOVIAS3DPh2[50] dataset. The best results are in bold.

compares the effects of these two modules on the effectiveness of the proposed stereo video deblurring method. The mPAM improves the PSNR by 0.59 dB.

Stereo consistency

To calculate the consistency between deblurred and original stereo video frames, we further investigate the end-point error (EPE) using the Euclidean distance among the two disparities (in original and deblurred videos), we estimate the disparity between the stereo frames of the reference videos with the approach proposed in Hirschmuller et al.[59] before calculating the disparity of the deblurred video frames. We calculate the EPE between two disparity values as the Euclidean distance between them. The results are shown in Table 3. The average EPE of our method is 0.7196 on the Stereo Blur dataset. In comparison, DAVANet[42] receives the average EPE of 0.7380 on the same dataset. Our method maintains better stereo consistency in the deblurring results.

Qualitative results

Figure 8 demonstrates the qualitative performance of our method on some stereo video frames from the Stereo Blur dataset. We compare our results with two 2D video deblurring methods (Son et al.[35], Pan et al.[34]), and one stereo image deblurring method, namely DAVANet[42]). We selected six video frames for this comparison, and in most of them, our method qualitatively outperforms the other methods. This figure shows that our approach efficiently uses the data from the neighboring frames. When the frame is blurry, the nearby frames help to deblur the middle frame. Additionally, Figure 7 illustrates the performance of the proposed method in stereo settings on the Stereo Blur dataset. The first row depicts the left frame, while the second row shows the right frame of a sample test video.

Ablation studies

We perform an extensive ablation study on the Stereo Blur[42] dataset to analyze the impact of various components within our model. This involves systematically removing specific modules (i.e., Transformer, mPAM module, Decoder, and a consecutive number of frames) and evaluating the resulting effect on the model's performance (PSNR and SSIM) as shown in Table 4 and Fig. 9. We refer to the architecture in Fig. 1 for this analysis.

Effect of the transformer

We remove the Transformer from both the left and right channels to see the effectiveness of our model performance. Since the left (PWC-Net) and right (CRB) sides of the Transformer in Fig. 1 contain 3D and 2D CNNs, respectively, we cannot directly remove the Transformer. Let's say the output of the PWC-Net is a 5 dimensional tensor with size $(Batch-size, N-frames, N-channel, W, H)$. We reshape the tensor to make the input tensor with 4 dimensions $(Batch-size, N-frames \times N-channel, W, H)$, then apply it to the CNN network. The results are shown in the first row of Table 4. The Transformer has a notable influence on the model efficiency, and the model PSNR decreases from 34.06 to 30.13 after removing the Transformer from the left and right channels. As shown in Fig. 9, when we disable the Transformer module, the model performance drops essentially.

Effect of the mPAM

We further investigate the effect of the cross-view information and deblur the left and right frames independently without considering the mPAM module. The model performs similarly to two models trained separately

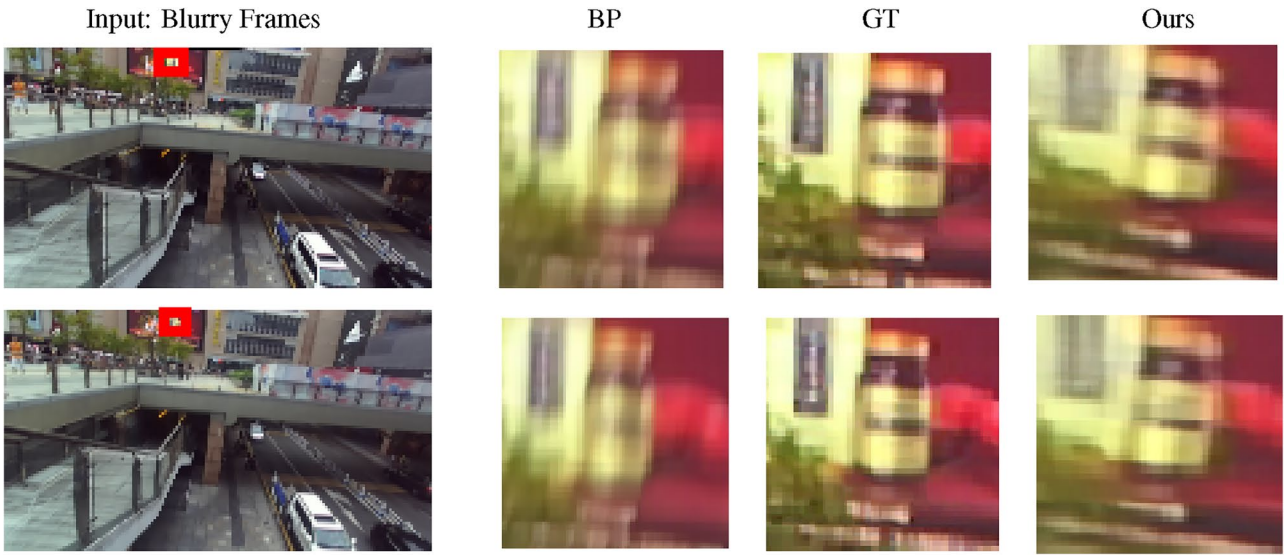

Figure 7. Qualitative performance of the proposed method on the Stereo Blur[42] dataset. The first row displays the left frame, and the second row displays the right frame of a sample test video. The BP and GT refer to the selected Blurry Part (BP) and Ground Truth (GT) of the video frame. .

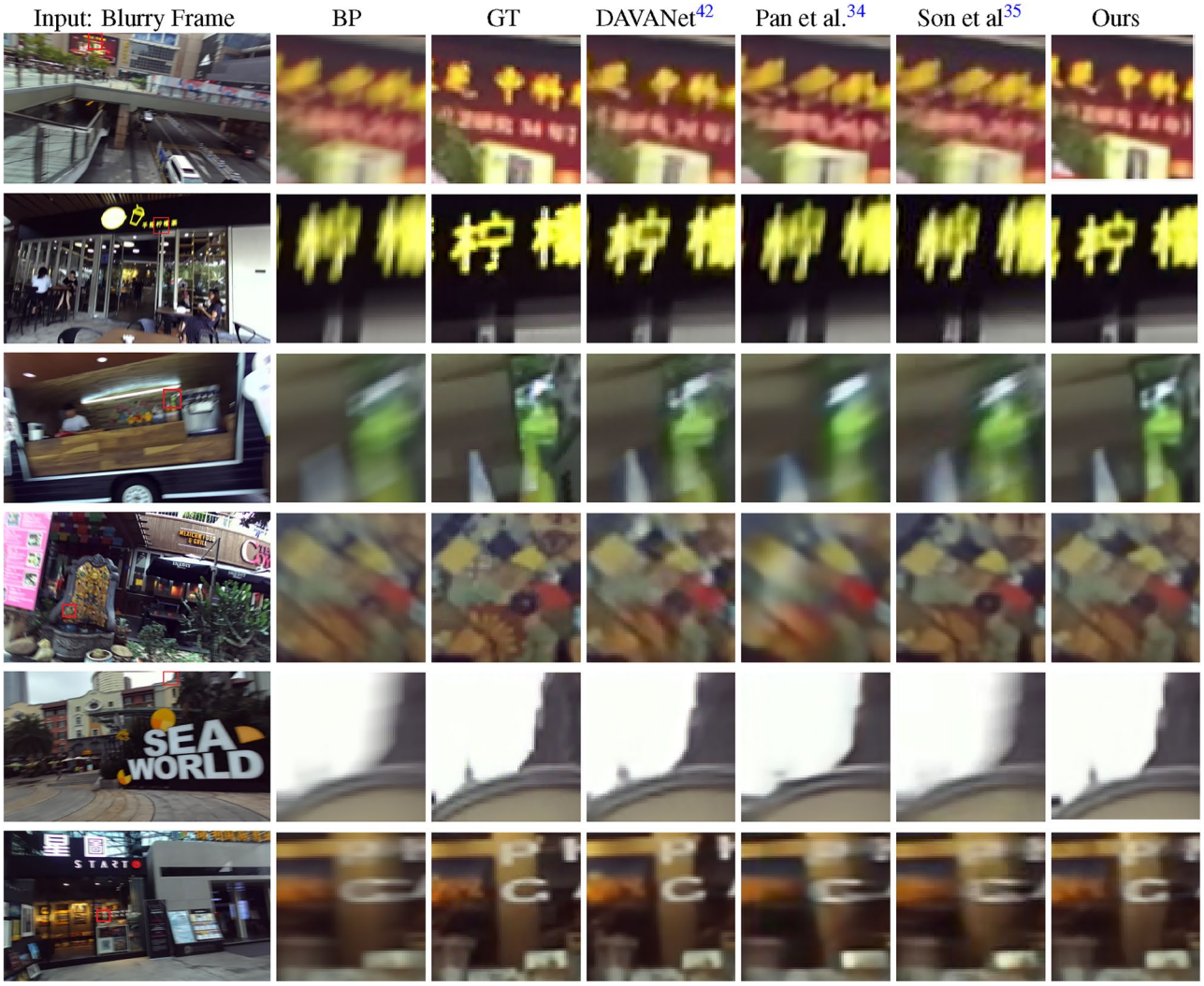

Figure 8. Qualitative performance comparison with state-of-the-art methods on different indoor and outdoor video frames in the Stereo Blur[42] dataset. The BP and GT refer to the selected Blurry Part (BP) and Ground Truth (GT) of the video frames.

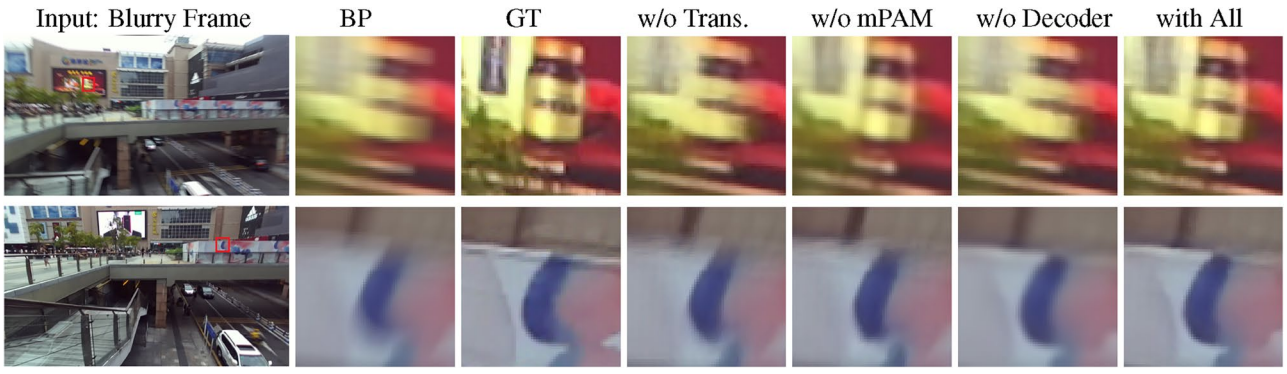

Figure 9. Qualitative performance comparison of our method, with and without different contributing modules, on two video frames on Stereo Blur[42] dataset. BP and GT refer to the selected Blurry Part of the frame and Ground Truth frame, respectively.

Stereo-based methods	Params (M)	EPE
DAVANet[42]	33.19	0.7380
UNet-Deblur[43]	30.56	0.7584
Ours	34.06	0.7196

Table 3. Stereo consistency. The average EPE of the proposed method against the stereo-based methods.

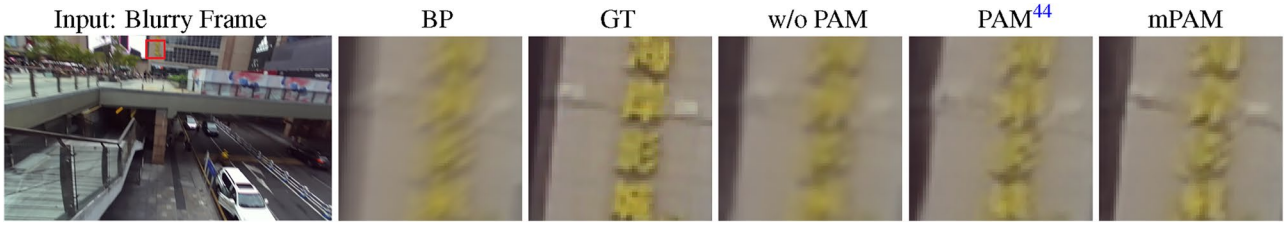

Figure 10. Effect of different PAM configurations in the overall performance of the proposed method on a video frame from Stereo Blur[42] dataset: w/o PAM: without PAM in our model, PAM[44], mPAM: modified PAM. The BP and GT refer to the selected Blurry Part (BP) and Ground Truth (GT) of the video frame.

Model settings	Params (M)	PSNR	SSIM
w/o Trans.	21.4M	30.13	0.9359
w/o mPAM	36.4M	30.39	0.9378
w/o Decoder	16.7M	31.44	0.9461
With all modules	38.4M	**34.06**	**0.9742**

Table 4. Performance comparison with (w) and without (w/o) contributing modules on Stereo Blur[42] dataset. Significant values are in bold.

Model settings	Params (M)	PSNR	SSIM
PAM[44]	38.1M	33.47	0.9568
mPAM	38.4M	34.06	0.9742

Table 5. Comparison the performance between the PAM[44] and the mPAM on Stereo Blur[42] dataset.

without using the cross-view information. The result of this change is illustrated in the second row of Table 4. Even without using the cross-view information, the proposed method outperforms image-based methods of Whyte[58], Sun[24], Gong[25], Nah[19], and Kupyn[57]. However, DAVANet[42], which uses the cross-view information efficiently, performs better than the proposed method without the mPAM module. Our model effectively uses the cross-view information, and the features from the other view help with further deblurring. The quantitative and qualitative influence of the mPAM module is shown in Table 5 and Fig. 10, respectively.

Effect of decoder

Since the output of the mPAM module has 32 filters, we use a 2D convolution after the mPAM to create a 3 channel output to add to the blurry input frames. We remove the convolutional decoder and add the output of the mPAM module to the blurry middle frame to create the deblurred output frames. The result is shown in the third row of Table 4, which shows the importance of the decoder module. This table shows that the decoder

N_frames	PSNR	SSIM
3	32.29	0.9507
5	34.06	0.9742
7	34.22	0.9777

Table 6. Impact of the number of input frames (*N_frames*) on the performance of the proposed model on the Stereo Blur[42] dataset. $N\_frames = 5$ demonstrates a favorable trade-off between performance and complexity.

module includes 21.7 million of parameters, a high number compared to other parts of our model. In the future, we will work on reducing the complexity of the decoder module.

Effect of consecutive frames numbers
In Sect. Quantitative results , we highlighted the use of a sequence consisting of 5 consecutive frames in our experiments. Here, we investigate how altering the number of input frames affects the performance of our model. Table 6 presents a comparative analysis across different frame counts, specifically $N\_frames = 3$, 5, and 7. The results demonstrate that selecting $N\_frames = 5$ yields optimal performance for stereo video deblurring. Notably, our proposed method exhibits sub-optimal performance with $N\_frames = 3$, while only marginal improvements are observed with $N\_frames = 7$. Therefore, choosing $N\_frames = 5$ strikes a favorable balance between performance and complexity.

Limitations
The increased number of model parameters in the proposed technique compared to image and 2D video deblurring methods is one of its drawbacks. As shown in Table 1, our model includes 38.4 million parameters, compared to 19.9 million for UNet-Deblur[43], 32.4 million for Pan et al.[34], 21.02 million for Son et al.[35], and 8.68 million for DAVANet[42]. This increase in parameter count is logical given that our proposed method addresses video deblurring with additional stereo-related information compared to 2D image-based and video-based methods. The inclusion of the temporal dimension inherently results in a model with higher complexity, such as using 3D convolutions instead of 2D. However, in the future, we aim to refine the modules of the overall architecture to make it more lightweight.

Conclusions
This paper proposed a new model for deblurring stereoscopic videos, marking the first Transformer-based stereo video deblurring method. We design its self-attention and feed-forward layers specifically for stereoscopic video deblurring. Additionally, we develop a method for fusing stereo information to enhance deblurring further. Our approach utilizes neighboring frames of a monocular view and corresponding stereo view to deblur the target frame. Extensive experiments demonstrate that our proposed approach outperforms both image and video-based deblurring methods on two benchmark datasets. In future work, we plan to optimize different parts of the proposed model to reduce complexity. Specifically, we aim to redesign the decoder to achieve comparable performance with fewer parameters. Additionally, we intend to refine the motion compensation module to focus more on the motion or salient parts of stereo videos.

Data availability
The source code for this work is available upon request to corresponding author(s).

Received: 5 June 2023; Accepted: 3 June 2024
Published online: 21 June 2024

References
1. Cao, J., Li, Y., Zhang, K. & Van Gool, L. Video super-resolution transformer. arXiv preprint arXiv:2106.06847 (2021).
2. Li, Y., Zhang, K., Cao, J., Timofte, R. & Van Gool, L. Localvit: Bringing locality to vision transformers. arXiv preprint arXiv:2104.05707 (2021).
3. Liu, C. & Sun, D. A Bayesian approach to adaptive video super resolution. In *CVPR 2011*, 209–216 (IEEE, 2011).
4. Baker, S. *et al.* A database and evaluation methodology for optical flow. *Int. J. Comput. Vis.* **92**, 1–31 (2011).
5. Xue, T., Chen, B., Wu, J., Wei, D. & Freeman, W. T. Video enhancement with task-oriented flow. *Int. J. Comput. Vis.* **127**, 1106–1125 (2019).
6. Fanous, M. J. & Popescu, G. Ganscan: continuous scanning microscopy using deep learning deblurring. *Light Sci. Appl.* **11**, 265 (2022).
7. Pan, L., Dai, Y., Liu, M. & Porikli, F. Simultaneous stereo video deblurring and scene flow estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4382–4391 (2017).
8. Sellent, A., Rother, C. & Roth, S. Stereo video deblurring. In *European Conference on Computer Vision*, 558–575 (Springer, 2016).
9. Imani, H., Islam, M. B. & Wong, L.-K. A new dataset and transformer for stereoscopic video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 706–715 (2022).
10. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020).
11. Sun, D., Yang, X., Liu, M.-Y. & Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8934–8943 (2018).
12. Xu, L., Zheng, S. & Jia, J. Unnatural l0 sparse representation for natural image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1107–1114 (2013).
13. Pan, J., Sun, D., Pfister, H. & Yang, M.-H. Blind image deblurring using dark channel prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1628–1636 (2016).
14. Li, L. *et al.* Blind image deblurring via deep discriminative priors. *Int. J. Comput. Vis.* **127**, 1025–1043 (2019).
15. Lee, D., Park, H., Park, I. K. & Lee, K. M. Joint blind motion deblurring and depth estimation of light field. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 288–303 (2018).
16. Park, H. & Mu Lee, K. Joint estimation of camera pose, depth, deblurring, and super-resolution from a blurred image sequence. In *Proceedings of the IEEE International Conference on Computer Vision*, 4613–4621 (2017).
17. Hu, Z., Xu, L. & Yang, M.-H. Joint depth estimation and camera shake removal from single blurry image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2893–2900 (2014).
18. Zoran, D. & Weiss, Y. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, 479–486 (IEEE, 2011).
19. Nah, S., Hyun Kim, T. & Mu Lee, K. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3883–3891 (2017).

20. Tao, X., Gao, H., Shen, X., Wang, J. & Jia, J. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8174–8182 (2018).
21. Zhang, J. *et al.* Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2521–2529 (2018).
22. Liang, C.-H., Chen, Y.-A., Liu, Y.-C. & Hsu, W. H. Raw image deblurring. *IEEE Trans. Multim.* **24**, 61–72 (2020).
23. Honarvar Shakibaei Asli, B., Zhao, Y. & Erkoyuncu, J. A. Motion blur invariant for estimating motion parameters of medical ultrasound images. *Sci. Reports* **11**, 14312 (2021).
24. Sun, J., Cao, W., Xu, Z. & Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 769–777 (2015).
25. Gong, D. *et al.* From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2319–2328 (2017).
26. Noroozi, M., Chandramouli, P. & Favaro, P. Motion deblurring in the wild. In *German conference on pattern recognition*, 65–77 (Springer, 2017).
27. Chang, M., Yang, C., Feng, H., Xu, Z. & Li, Q. Beyond camera motion blur removing: how to handle outliers in deblurring. *IEEE Trans. Comput. Imag.* **7**, 463–474 (2021).
28. Li, L. *et al.* Dynamic scene deblurring by depth guided model. *IEEE Trans Image Process* **29**, 5273–5288 (2020).
29. Zamir, S. W. *et al.* Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 5728–5739 (2022).
30. Kong, L., Dong, J., Li, M., Ge, J. & Pan, J. Efficient frequency domain-based transformers for high-quality image deblurring. arXiv preprint arXiv:2211.12250 (2022).
31. Delbracio, M. & Sapiro, G. Hand-held video deblurring via efficient fourier aggregation. *IEEE Trans. Comput. Imaging* **1**, 270–283 (2015).
32. Su, S. *et al.* Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1279–1288 (2017).
33. Zhang, K. *et al.* Adversarial spatio-temporal learning for video deblurring. *IEEE Trans. Image Process.* **28**, 291–301 (2018).
34. Pan, J., Bai, H. & Tang, J. Cascaded deep video deblurring using temporal sharpness prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3043–3051 (2020).
35. Son, H., Lee, J., Lee, J., Cho, S. & Lee, S. Recurrent video deblurring with blur-invariant motion estimation and pixel volumes. *ACM Trans. Graphics (TOG)* **40**, 1–18 (2021).
36. Zhang, X., Jiang, R., Wang, T. & Wang, J. Recursive neural network for video deblurring. *IEEE Trans. Circuits Syst. Video Technol.* **31**, 3025–3036 (2020).
37. Wang, X. Z. T., Jiang, R., Zhao, L. & Xu, Y. Multi-attention convolutional neural network for video deblurring. *IEEE Trans. Circuits Syst. Video Technol.* (2021).
38. Yan, Y., Wu, Q., Xu, B., Zhang, J. & Ren, W. Vdflow: Joint learning for optical flow and video deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 872–873 (2020).
39. Dosovitskiy, A. *et al.* Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, 2758–2766 (2015).
40. Chen, H., Teng, M., Shi, B., Wang, Y. & Huang, T. A residual learning approach to deblur and generate high frame rate video with an event camera. *IEEE Trans. Multim.* (2022).
41. Xu, L. & Jia, J. Depth-aware motion deblurring. In *2012 IEEE International Conference on Computational Photography (ICCP)*, 1–8 (IEEE, 2012).
42. Zhou, S. *et al.* Davanet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10996–11005 (2019).
43. Imani, H. & Islam, M. B. Towards stereoscopic video deblurring using deep convolutional networks. In *International Symposium on Visual Computing*, 337–348 (Springer, 2021).
44. Wang, L. *et al.* Learning parallax attention for stereo image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12250–12259 (2019).
45. Jeny, A. A., Islam, M. B. & Aydin, T. Deeppynet: A deep feature pyramid network for optical flow estimation. In *2021 36th International Conference on Image and Vision Computing New Zealand (IVCNZ)*, 1–6 (IEEE, 2021).
46. Horn, B. K. & Schunck, B. G. Determining optical flow. *Artif. Intell.* **17**, 185–203 (1981).
47. Vaswani, A. *et al.* Attention is all you need. *Adv. Neural Inf. Process. Syst.*, 5998–6008 (2017).
48. Wang, Y. *et al.* End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8741–8750 (2021).
49. Ranjan, A. & Black, M. J. Optical flow estimation using a spatial pyramid network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4161–4170 (2017).
50. Appina, B., Dendi, S. V. R., Manasa, K., Channappayya, S. S. & Bovik, A. C. Study of subjective quality and objective blind quality prediction of stereoscopic videos. *IEEE Trans. Image Process.* **28**, 5027–5040 (2019).
51. Zed 2 - AI Stereo Camera. . Stereolabs. https://www.stereolabs.com/zed-2/
52. Sundaram, N., Brox, T. & Keutzer, K. Dense point trajectories by gpu-accelerated large displacement optical flow. In *European conference on computer vision*, 438–451 (Springer, 2010).
53. Imani, H., Islam, M. B. & Arica, N. Three-stream 3d deep cnn for no-reference stereoscopic video quality assessment. *Intell. Syst. Appl.* **13**, 200059 (2022).
54. Imani, H., Zaim, S., Islam, M. B. & Junayed, M. S. Stereoscopic video quality assessment using modified parallax attention module. In *Digitizing Production Systems: Selected Papers from ISPR2021, October 07-09, 2021 Online, Turkey*, 39–50 (Springer, 2022).
55. Imani, H., Islam, M. B., Junayed, M. S., Aydin, T. & Arica, N. Stereoscopic video quality measurement with fine-tuning 3d resnets. *Multim. Tools Appl.* 1–21 (2022).
56. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014).
57. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D. & Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 8183–8192 (2018).
58. Whyte, O., Sivic, J., Zisserman, A. & Ponce, J. Non-uniform deblurring for shaken images. *Int. J. Comput. Vis.* **98**, 168–186 (2012).
59. Hirschmuller, H. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.* **30**, 328–341 (2007).

Acknowledgements

This work was supported in part by the Scientific and Technological Research Council of Turkey (TUBITAK) through the 2232 Outstanding International Researchers Program under Project No. 118C301.

Author contributions

H.I.: Experiment, methodology development, result analysis, initial drafting. M.B.I.: Conceptualization, investigation, supervision, reviewing, and editing of the manuscript. M.S.J.: Experiment, initial drafting. M.A.R.A.: Analysing, Reviewing, and editing the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.B.I. or M.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.