EasyChair Preprint

№ 6911

Towards Stereoscopic Video Deblurring Using Deep Convolutional Networks

Hassan Imani and Md Baharul Islam

October 21, 2021

Towards Stereoscopic Video Deblurring Using Deep Convolutional Networks

Hassan Imani, and Md Baharul Islam

Department of Computer Engineering, Bahcesehir University, Turkey
hassan.imani1987@gmail.com, bislam.eng@gmail.com

Abstract. These days stereoscopic cameras are commonly used in daily life, such as the new smartphones and emerging technologies. The quality of the stereo video can be affected by various factors (e.g., blur artifact due to camera/object motion). For solving this issue, several methods are proposed for monocular deblurring, and there are some limited proposed works for stereo content deblurring. This paper presents a novel stereoscopic video deblurring model considering the consecutive left and right video frames. To compensate for the motion in stereoscopic video, we feed consecutive frames from the previous and next frames to the 3D CNN networks, which can help for further deblurring. Also, our proposed model uses the stereoscopic other view information to help for deblurring. Specifically, to deblur the stereo frames, our model takes the left and right stereoscopic frames and some neighboring left and right frames as the inputs. Then, after compensation for the transformation between consecutive frames, a 3D Convolutional Neural Network (CNN) is applied to the left and right batches of frames to extract their features. This model consists of the modified 3D U-Net networks. To aggregate the left and right features, the Parallax Attention Module (PAM) is modified to fuse the left and right features and create the output deblurred frames. The experimental results on the recently proposed Stereo Blur dataset show that the proposed method can effectively deblur the blurry stereoscopic videos.

Keywords: Stereoscopic video · image deblurring · convolutional neural networks · motion · disparity · PAM

1 Introduction

Recently, many works of literature have been published for 2D image deblurring utilizing the power of deep learning. For example, [1, 2] used CNNs to estimate the blur kernels in the case of the non-uniform blur. Also, there were proposed many CNN-based models for image deblurring such as [3, 4]. Seungjun et al. [3] proposed a blind deblurring model for deblurring the videos with motion blur. This model is a multi-scale CNN-based network that tries to restore sharp frames. The method in [4] proposed based on the coarse-to-fine method, which is progressively restoring the sharp image on different resolutions to offer a technique that is less complex than the previous methods, and its results are better

than the previous ones. The proposed model is also a multi-scale network. Zhang et al. [5] proposed a method that is designed to cope with the spatially variant blur when there is a camera motion. They used three CNNs and an RNN.

Motion plays an essential role in video processing tasks [6]. When the aim is to deblur a video, motion information can be used. It is because most of the blur in video frames is due to motion. This motion can be the camera motion or the movement of the objects in the video causing the blur. Most of the video-related methods, firstly, calculate the motion between the consecutive frames, and secondly, use a processing step including the transformation to the frames [7, 8]. As a result, the accuracy of the motion estimation directly influences the performance of the overall method. However, accurate motion estimation is complex and slow [6]. Most methods assume that the brightness is constant, but this assumption may not be valid because of the inevitable change in lighting and pose and the motion blur and occlusion. Besides, most of the motion estimation methods solve an optimization problem, making the motion estimation slow.

Because of the change in the disparity and wanted or unwanted movement of the camera, calculation and compensation for the spatial blur using the spatial data from one view seems complicated. The amount of the information appears to be insufficient. However, some of the deep learning-based models, such as [9], reached an acceptable performance for 2D image deblurring. However, their performance drops when the amount of the blur is not scattered in the image uniformly. For stereoscopic video deblurring, [10] utilized the information from the left and right views for deblurring, where a coarse depth or piecewise rigid 3D scene flow is utilized to estimate blur kernels in a hierarchical or iterative framework. However, they have a complicated optimization method, and it makes these methods challenging to use everywhere.

Some literature used the stereo disparity and video motion for stereoscopic image and video deblurring. The authors in [11] estimated the depth layers and layer-specific point spread functions and used them to deblur the images. They firstly calculated the disparity utilizing the information from the left and right blur images, and then they proposed a region tree scheme to calculate the point spread functions. Sellent et al. [12] considered scene flow and stereo video deblurring as a mutual task. They used local homographs to create blur kernels using scene flow estimation. They used a weighting method in the boundaries of the objects with a motion to spot the degradations accurately. In this work, the scene flow and deblurring are considered separately, and the previously computed scene flow is used. Recently, DAVANet [13] is proposed, which consists of three parts: an encoder-decoder network architecture, a disparity estimation network, and a fusion model to combine the two models and create the deblurred images. They also proposed a new dataset named Stereo Blur dataset. We used this dataset for the experiments in this paper.

Motivation. Our motivation for proposing a new model for stereoscopic video deblurring is based on two facts: (1) Aggregating the information from the consecutive frames of one view can help to spot the artifacts in pixels of the middle frame using the motion information between the successive frames. In other

Fig. 1. Five left frames from the Stereo Blur Dataset [13]. As can be seen from these consecutive frames, some blur regions in the target middle frame are not blurred in the surrounding frames, and these frames can help deblur the middle frame.

words, because of the small motion between the limited number of consecutive frames, for deblurring one frame of a video, the neighboring frames can help to deblur the target frame. For example, as shown in Figure 1, suppose that we want to deblur the middle frame. Some regions are blurred in the middle frame but not in the neighboring 1, 2, 4, or 5 frames. After compensating for motion, we can use them to deblur the middle target frame. (ii) In stereo vision, two views from one scene are available. We believe that with the disparity information, the corresponding pixels in one view can help remove the blur from the other view. In most cases, the blur in the left and right frames is not the same.

Contributions. This paper proposes a new deep learning-based stereoscopic video deblurring model to cope with the motion blur in dynamic scenes. This model uses the neighboring frames and the information from the other stereo view to deblur one frame. Specifically, we first find the keypoints and descriptors for the middle frame and the neighbouring frames, and then match the features among the two images using the Brute Force method. Then, we calculate the homography matrix to transform the neighbouring frame to the middle frame to consider for the camera rotation and translation. Then, we extract the features from the main frame and the transformed neighboring frames using some modified 2D U-Net [14] networks that using 3D CNNs. Finally, we aggregate the extracted features from the left and right views using a modified PAM [15] model and create the deblurred middle frame. The main contributions of this paper are as follows:

1. We propose a new stereoscopic video deblurring model. The proposed model uses the information from the other stereo view and the neighboring frames to deblur the middle frame.
2. To fuse the left and right videos features, the PAM module is modified to make it suitable for aggregating features from the stereo videos.
3. The 2D U-Net model is modified to make it suitable for extracting features from the 3D input as the batch of the consecutive frames.

2 Proposed Method

The architecture of the proposed method is shown in Figure 2. As can be seen from this Figure, for each view, we select the middle frame from the consecutive

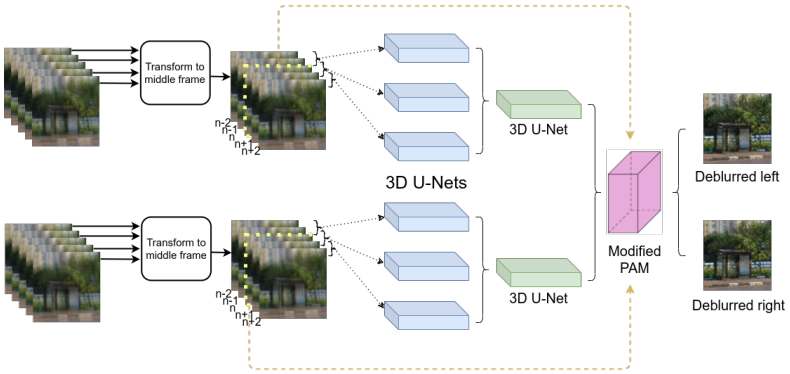

Fig. 2. Overview of the proposed stereoscopic video deblurring architecture. In this architecture, firstly, 4 left and right neighboring frames are transformed to the corresponding middle frames. Then, eight 3D U-Net models are used to extract features from the left and right neighboring frames, and a modified PAM module is used to fuse the left and right features and create the deblurred frames.

frames as the frame that we want to deblur. Let us say frame number n is selected from the left and right videos as the middle frames that we want to deblur. Two previous and two next frames $(n-1, n-2, n+1, n+2)$ are also selected. The motion between the middle frame and these four frames is calculated, and based on this motion, the frames $(n-1, n-2, n+1, n+2)$ are transposed to the middle frame n to have different representations of the target frame. Then, the resulting five frames are used as input to the modified U-Net networks to extract their features. Three modified U-Net networks are used to extract features from the five frames from each view. In [16, 17] the authors discussed that how multi-scale networks similar to the 2D U-Net can help to learn the misalignments. Besides, in [18] the experimental results show that the cascaded framework can further improve the ability to handle the movements which are used for video denoising. Therefore, based on their impressive results for different tasks, we designed our model based on the cascaded architecture. Frames n, $n-1$, and $n-2$ to the first, frames $n+1, n$, and n-1 to the second, and frames $n, n+1$, and $n+2$ are feed to the third modified U-Net networks. The layers of the modified U-Net networks use 3D convolutional layers and are suitable to accept a stack of frames. The output of the U-Net in the first stage is used as input to another U-Net. The second stage U-Net is applied to the output features of the first stage U-Nets.

After extracting features from the left and right batches of frames, they are used as input to the modified PAM module. The modified PAM module is used to fuse the left and right features and consider the disparity between the left and right views. Finally, the signals from the output of the modified PAM module are fed to a convolutional layer to create the deblurred version of the input middle left and right frames. We added two 2D convolutional layers at the output of the modified PAM module, each followed by a ReLU activation function. In between

the 3D convolutional layers, we added a Batch Normalization (BN) layer to standardize the input features. The final convolutional layer's output has just one filter to make the whole network have an output with the same size as the middle input frame.

As can be seen from Figure 3, the neighboring frames are transformed to the middle frame before applying to the model. As shown in this figure, we first convert both the middle frame (number 3) and the neighbouring frame to a grayscale image. Then, we use the keypoint detector and descriptor method named Oriented fast and Rotated BRIEF (ORB) [19] to extract their descriptors. In this method, a binary descriptor based on BRIEF [20] which is rotation invariant and resistant to noise, is used. ORB is two times faster than SIFT [21]. To match the extracted features, we use the Brute-Force matcher [22]. It takes the descriptor of one feature in the first set and matches it with all other features in the second set using the distance between them. The closest feature is identified as the matching feature. Finally, the homography matrix calculated from the matched features is used to transform the colored frame number 2 to the middle frame number 3. This process is used to transform all other three frames to the middle frame to make them ready to use as the inputs to our deblurring model.

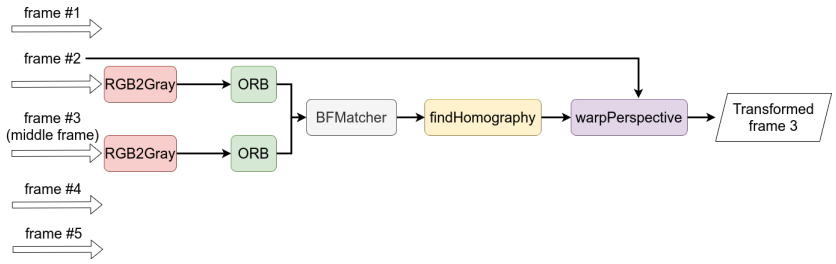

Fig. 3. Transformation of the neighbouring frames to the middle frame. Here we transform the neighbouring frame number 2 to the middle frame number 3. After RGB to grayscale image conversion, we apply the Oriented Rotated BRIEF (ORB) to each frame. Then, using the Brute-Force matcher, we match the features among the two frames and find the homography matrix (findHomography). Finally, the homography matrix is used to transform the colored frame number 2 to the middle frame 3.

2.1 3D U-Net Architecture

The original U-Net [14] model includes one contracting path to catch the context and one symmetric expanding path that allows accurate localization and uses 2D convolutions and other 2D max-pooling which its input is a 2D image. To utilize the U-Net architecture when the inputs are multiple images, we modified the original U-Net architecture. The architecture of the modified U-Net is depicted in Figure 4. In this Figure, the contracting path consists of 3D convolutional layers.

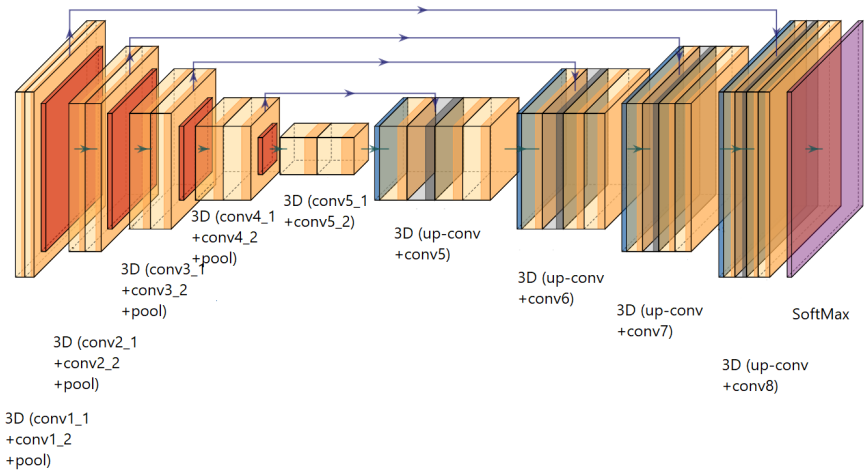

Fig. 4. The architecture of the modified U-Net. We used 3D CNNs and 3D max-pooling layers to make the modified U-Net suitable for the input stack of the frames. Every 3D convolution is followed by a ReLU.

This path includes two $5 \times 5 \times 5$ convolutions, followed by a ReLU activation function. A 3D $2 \times 2 \times 2$ max-pooling with a stride of 2 is added to downsample the features. In the expansive path, the contracting path is done in an inverse manner. The feature maps are up-sampled, and a $2 \times 2 \times 2$ convolution is used to reduce the number of feature channels to 50%. Then, we add a concatenation layer and two $5 \times 5 \times 5$ convolutions, each followed by a ReLU activation function. This architecture is repeated for each of the 3D U-Nets in our proposed model in Figure 2.

In the modified U-Net architecture, we changed the 2D convolutions to 3D and used 3D max-pooling. These changes can help to compensate for motion in the videos too. In the 3D convolutions, consecutive frames instead of just one frame are included. Therefore, using 3D convolution and 3D max-pooling has the advantage of considering the motion in the video implicitly.

2.2 Modified PAM Architecture

Based on the self-attention mechanisms [23, 24], Wang et al. in [15] proposed 2D PAM to calculate global correspondence in stereoscopic images. PAM effectively fuses the data from the stereo image pairs. The original PAM architecture is modified to make it suitable for our inputs which have the time dimension added. We also converted the residual block to 3D. The architecture of the modified PAM is shown in Figure 5. "L" and "R" show the left and right features, which are input to the modified PAM module. They are fed to the 3D residual blocks. The output is fed into a $1 \times 1 \times 1$ 3D convolutional layer. Then, batch-wised matrix multiplication is applied, and the resulted feature maps are passed to a

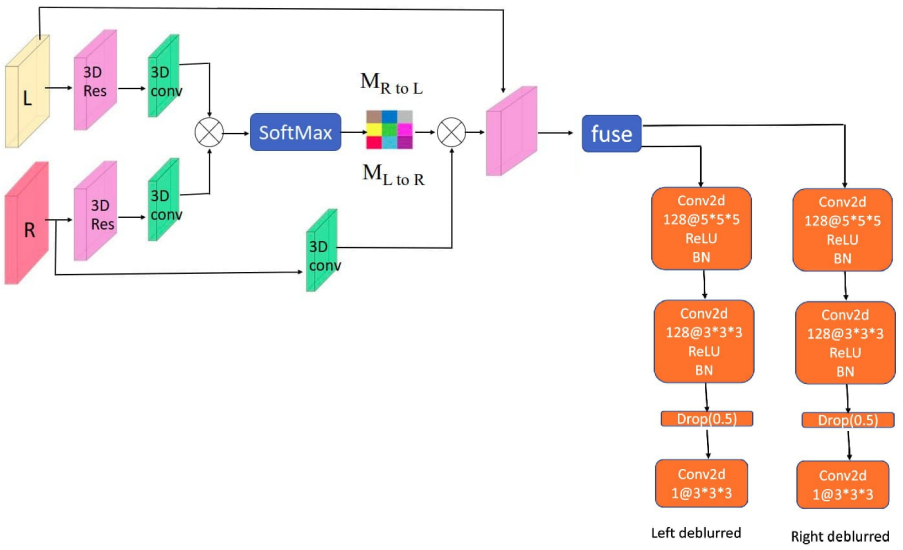

Fig. 5. The modified PAM module. The input features are fed to a transition residual block, and the output is the integration of the input features. 2D convolutions are converted to 3D.

SoftMax layer to create the parallax attention maps $M_{R\ to\ L}$ and $M_{L\ to\ R}$. Then, the weighted sum of features at all disparity values is fused with their input left features. Finally, the features are fused to create the output features for the left feature map. For the fusion of the right features, the process is the same, and just the left feature fusion is shown in Figure 5.

We added three convolutional layers followed by a ReLU activation and batch normalization to make rich features suitable for deblurring. The first 2D convolution has 128 filters and followed by a ReLU activation function and a batch normalization layer. The size of the kernel for this convolution is 5×5. The second convolutional layer is similar to the first one with a kernel size of 3×3. Next, a dropout layer with a rate=0.5 is applied. Using a dropout layer is one way to avoid overfitting. This layer forces the activation values of random neurons to zero. This layer help to design larger models with more nodes per layer, since it probabilistically decreases the capacity of the network and prevents the model from overfitting. The last layer is a convolutional layer with 64 filters and a kernel size of 3×3.

3 Dataset and Experimental Setup

Because of the unavailability of the datasets for stereoscopic video deblurring, we trained and tested our model on just one dataset.

Fig. 6. Sample left frames from the Stereo Blur dataset. This dataset contains 20,637 frames from 135 stereo videos.

3.1 Dataset

The Stereo Blur dataset is proposed in [13]. This dataset includes comprehensive indoor and outdoor scenarios. In the indoor videos, there are videos from objects and persons with small disparities. The outdoor videos consist of people, vehicles, boats, and natural scenes. Besides, the dataset is variegated using several factors such as illumination and weather changes. The dataset is further extended to cover various motion scenarios using three different types of photography fashions: handheld, fixed, and onboard shots.

For the creation of the dataset, the authors used the ZED stereo camera [25], which has a frame rate of 60 Frames Per Second (FPS). They expanded the final video frame rate to 480 FPS using a frame interpolation method in [26]. Both the left and right views of the stereo video are with the same configurations. Moreover, this dataset consists of the disparity between the left and right views and the mask map for rejecting the inaccurate values in disparity ground truth and occluded parts of the images, which are calculated by bidirectional consistency check [27]. Overall, the dataset contains sequences of 135 stereo videos from dynamic scenes. The videos are converted to frames, and a total of 20,637 blurred and reference stereo frame pairs together with their corresponding disparities are included in the dataset. The dataset is split into 98 training and 37 testing sequences.

3.2 Experimental Setup

We train the proposed stereo deblurring model on the Stereo Blur dataset. Firstly, we center-cropped all the frames in the dataset with a size of 256×256 and created another dataset. In the created dataset, for each middle left and right frame (number n), we created a folder containing the middle frame n and four other frames $(n-2, n-1, n+1, n+2)$. Frame number n folder contains 5 left, 5 right, and 2 ground truth frames called in the training process.

We use Stochastic Gradient Descent (SGD) with momentum to train our network. For doing faster convergence, SGD drives gradients in the right direction. The random division of the dataset is done in [13] and that division is used here.

Table 1. Performance evaluation of the proposed stereo deblur model and comparison with the image-based deblurring methods on Stereo Blur Dataset, in terms of PSNR, SSIM, running time, and parameter number. A "-" indicates that the result is not available. The best results are in **bold**.

Image-based methods	PSNR	SSIM	Time (sec)	Params (M)
Whyte [28]	24.48	0.8410	700	-
Sun [1]	26.13	0.8830	1200	**7.26**
Gong [2]	26.51	0.8902	1500	10.29
Nah [3]	30.35	0.9294	4.78	11.71
Kupyn [9]	27.81	0.8895	**0.22**	11.38
Zhang [5]	30.46	0.9367	1.40	9.22
Tao [4]	31.65	0.9479	2.52	8.06
DAVANet [13]	**33.19**	**0.9586**	0.31/pair	8.68
Video-based method (**ours**)	30.56	0.9221	0.57/pair	19.9

The batch size for training is selected as 1 on NVIDIA GTX 2080 ti. The proposed model is implemented and trained using PyTorch 1.8.1. The weight decay is 0.001, and the Nesterov momentum is 0.9. The initial learning rate is set to 0.0001 and divided by 10 after the validation loss saturates. Mean Squared Error (MSE) is used as our loss function. We initialized the parameters randomly and trained them by the online error Back Propagation (BP) algorithm.

4 Results and Discussions

Due to the not publicly available stereoscopic video deblurring dataset, we conduct our experiments on the stereo blur dataset [13] which is initially developed for stereoscopic image deblurring. This dataset contains stereo image sequences which are the frames of the stereo videos. Besides, to the best of our knowledge, no stereoscopic video deblurring method reported its results on this dataset. For this reason, we compare our results with the stereoscopic image deblurring methods and use the frame-based average for comparison.

We compare our stereo video deblurring results regarding Structural SIMilarity index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) to the deep learning-based and traditional methods. The networks of the CNN-based methods of [3, 9, 5, 4] are fine-tuned on the Stereo Blur dataset in [13]. Table 1 shows the results of the comparison between image and video-based deblurring methods. It can be seen that no stereo video deblurring method is reported their results on the Stereo Blur dataset. The table also shows that the proposed method perform better than the following image-based methods: Whyte et al. [28], Sun et al. [1], Gong et al. [2], Nah et al. [3], Kupyn et al. [9], and Zhang et al. [5]. Tao et al. [4] and DAVANet [13] are performing better than our method. Comparing the 2D or 3D image-based deblurring methods with our stereo video deblurring method may be misleading. To the best of knowledge, there are two stereo video deblurring method in the literature [12], [10]. These methods are not open-source, and they have not reported their results on the Stereo Blur dataset. The reason is

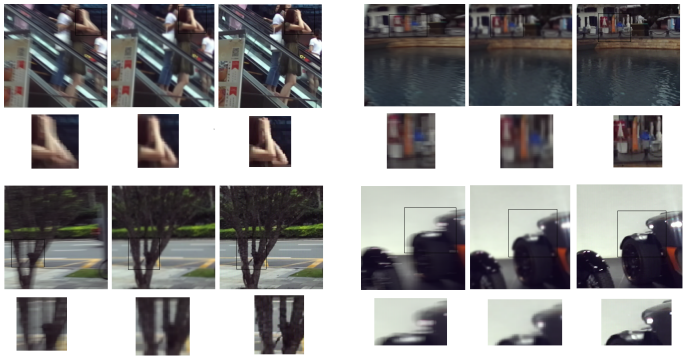

Fig. 7. Four sample qualitative test results from the Stereo Blur dataset. (Left to right) left blur frame, deblurred frame using our model, and ground truth.

that the Stereo Blur dataset is proposed after these methods. They did their experiments on some stereo videos that they generated for their usage. Sellent et al. [12] proposed a small-scale dataset which is stereo image-based. We could not use this dataset since it is not based on the consecutive frames which our algorithm needs at least 5 consecutive frames. Besides, it is small-scale and can not be used to train our deep learning-based model. The method in [10] created the blurred stereo images with synthetically blurring the stereo images in the KITTI [29] and Sellent et al. [12] datasets, and we could not compare our results with this method either. Therefore, we could not compare our method's performance with these methods.

Figure 7 shows the qualitative performance of our method on four test videos from the Stereo Blur dataset. As can be seen from this Figure, the motion blur is enhanced in most cases. It is the effect of using the neighboring frames for strengthening the central frame. It is because the motion blur is not appearing in all the consecutive frames. Consider a specific region in a frame. When there is a motion blur in the central frame, some neighboring frames are not blurred within that particular region. Therefore, fewer or non-blurred frames can be used to deblur the other frames, and our method effectively utilizes this information. This Figure indicates that the proposed method can deblur the stereo video with acceptable performance.

5 Conclusions and Future Works

This paper proposed a new stereoscopic video deblurring model that uses the stereo information and the neighboring frames to deblur the middle frame. Due to the slight motion in consecutive frames of a video, the past and previous frames carry information about the main frame. Instead of the blur frame itself, the information from the neighboring previous and past frames is used to deblur the target frame. The network architecture has two parts: the first part extracts

features from the middle frame and the neighboring frames of each view, and the second part fuses the left and right features and creates the deblurred frames. Quantitative and qualitative experimental results on the Stereo Blur dataset show that the proposed method can deblur the stereo videos with acceptable performance. From the experiments, it is noticeable that there is a need for a very big dataset to develop deep learning-based methods for stereoscopic video deblurring. In the future, we will train deep learning-based methods on the combination of several datasets, and create blurry frames synthetically and extend the dataset to avoid overfitting. Tao et al. [4] and DAVANet [13] are performing better than our method. In the future, we will adapt our model to use an encoder-decoder-like architecture similar to DAVANet [13], or we will design a new model based on the Transformers to get the state-of-the-art results.

References

1. J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 769–777.
2. D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "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*, 2017, pp. 2319–2328.
3. S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3883–3891.
4. X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8174–8182.
5. J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M.-H. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2521–2529.
6. T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.
7. C. Liu and D. Sun, "A bayesian approach to adaptive video super resolution," in *CVPR 2011*. IEEE, 2011, pp. 209–216.
8. S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, "A database and evaluation methodology for optical flow," *International journal of computer vision*, vol. 92, no. 1, pp. 1–31, 2011.
9. O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.
10. L. Pan, Y. Dai, M. Liu, and F. Porikli, "Simultaneous stereo video deblurring and scene flow estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4382–4391.
11. L. Xu and J. Jia, "Depth-aware motion deblurring," in *2012 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2012, pp. 1–8.

12. A. Sellent, C. Rother, and S. Roth, "Stereo video deblurring," in *European Conference on Computer Vision*. Springer, 2016, pp. 558–575.
13. S. Zhou, J. Zhang, W. Zuo, H. Xie, J. Pan, and J. S. Ren, "Davanet: Stereo deblurring with view aggregation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 996–11 005.
14. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
15. L. Wang, Y. Wang, Z. Liang, Z. Lin, J. Yang, W. An, and Y. Guo, "Learning parallax attention for stereo image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 250–12 259.
16. A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.
17. S. Wu, J. Xu, Y.-W. Tai, and C.-K. Tang, "Deep high dynamic range imaging with large foreground motions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 117–132.
18. M. Tassano, J. Delon, and T. Veit, "Fastdvdnet: Towards real-time deep video denoising without flow estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1354–1363.
19. E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.
20. M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*. Springer, 2010, pp. 778–792.
21. D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.
22. OpenCV modules,, "Accessed: 2017-04-19." [Online]. Available: http://docs.opencv.org/3.1.0.
23. H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.
24. J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3146–3154.
25. Capture the world in 3d. , "Stereolabs. (n.d.)." [Online]. Available: https://www.stereolabs.com/.
26. S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.
27. N. Sundaram, T. Brox, and K. Keutzer, "Dense point trajectories by gpu-accelerated large displacement optical flow," in *European conference on computer vision*. Springer, 2010, pp. 438–451.
28. O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International journal of computer vision*, vol. 98, no. 2, pp. 168–186, 2012.
29. A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.