Supplementary Materials to "A Differentiable Two-stage Alignment Scheme for Burst Image Reconstruction with Large Shift"

Shi Guo[1] Xi Yang[1] Jianqi Ma[1] Gaofeng Ren[2] Lei Zhang[1]
[1]The Hong Kong Polytechnic University; [2]DAMO Academy, Alibaba Group
{shiguo.guo,xxxxi.yang,jianqi.ma}@connect.polyu.hk,
gaof.ren@gmail.com, cslzhang@comp.polyu.edu.hk

In this supplementary file, we provide the following materials:

- Detailed network structure of the refined alignment module (referring to Section 3.3 in the main paper);
- Detailed network structure of the fusion module (referring to Section 3.4 in the main paper);
- Statistics of pixel shifts of REDS4 dataset (referring to Section 4.2 in the main paper);
- Ablation study on the impact of the two-stage alignment framework (referring to Section 3.1 in the main paper);
- Ablation study on the impact of the proposed DPBM (referring to Section 3.2 in the main paper);
- Ablation study on the impact of the interpolation loss (referring to Section 3.5 in the main paper);
- More visual comparisons of different methods on the Videezy4K dataset (referring to Section 4.2 in the main paper);
- More visual comparisons and user study of different methods on real-world burst images (referring to Section 4.2 in the main paper).

1. More Details of the Network Structure

The architecture of the refined alignment module and the fusion module are shown in Fig. 2. For the fusion model, we utilize the gate recurrent unit (GRU) to aggregate the forward/backward temporal information (Equ. 4 in the main paper). Here, we use $h_t = f_{gru}(F_t, h_{t-1})$ as an example to show the detailed GRU computing process:

$$\begin{aligned} z_t &= \sigma(w_z([w_{zh}h_{t-1}, w_{zf}F_t])), \\ r_t &= \sigma(w_r([w_{rh}h_{t-1}, w_{rf}F_t])), \\ \widetilde{h}_t &= \tanh(r_t \odot w_{hh}h_{t-1} + w_{hf}F_t), \\ h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h}_t, \end{aligned} \tag{1}$$

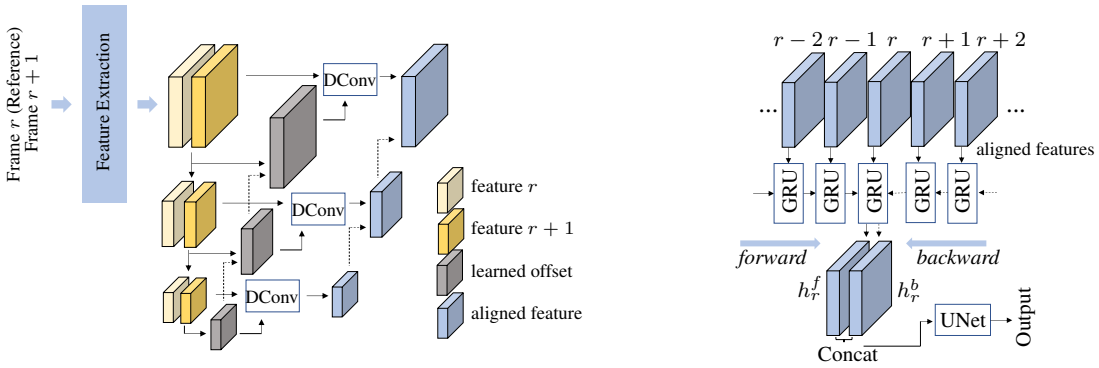
where σ is the activation function (*e.g.*, sigmoid), $\odot$ is the Hadamard product of matrix, and $[\cdot]$ is the concatenation operation. w_z, w_r, w_{zh}, w_{zf}, w_{rh}, w_{rf}, w_{hh} and w_{hf} are the weight matrices of the corresponding convolution layers.

2. Statistics of pixel shifts of REDS4 dataset

We compute the pixel shift between adjacent frames on REDS4 by the RAFT method [4], and plot the statistics in Fig. 2. We see that even for the 720p REDS4 images, the shifts can be very large (> 20). As shown in the right figure, the shift can be more than 70 pixels but our method can still handle it.

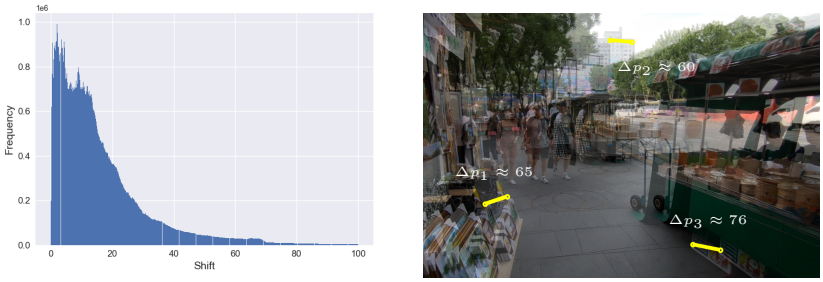
3. Ablation Studies

Two-stage Alignment vs. One-stage Deep Alignment. One interesting question is whether a single-stage but deeper feature alignment module can obtain comparable performance to our two-stage alignment method. To answer this question, we train a variant of ours(w/ RA) (the model which uses only the refined module) with a deeper alignment module, denoted as ours(w/ DRA). In the coarse alignment module, we set the search radius as 24 (in HR feature domain) and perform the alignment



(a) Refined alignment module (b) Fusion module

Figure 1. Architecture of the refined alignment module and the fusion module.



(a) Shift statistics (b) An example

Figure 2. (a) The histogram of shifts on the 720p REDS4 dataset. (b) An example of shift between frames in REDS4 dataset.

Table 1. Comparison of different variants of our method on a clip with small motion (*Clip 000*) and a clip with large motion (*Clip 020*) in the REDS4 dataset, and the average results on the whole REDS4.

	Clip 000	*Clip 020*	Avg
ours(w/ RA)	32.57/0.9190	32.69/0.9002	33.90/0.9077
ours(w/ DRA)	32.50/0.9183	33.16/0.9073	34.08/0.9092
ours(w/o $\mathcal{L}_{ip}$)	32.73/0.9140	34.07/0.9182	34.33/0.9128
ours(full)	**32.75/0.9174**	**34.17/0.9233**	**34.43/0.9178**

on the 1/4 scale. Therefore, we add 16 Conv layers in the offset estimation part of the refined alignment module in ours(w/ DRA) to ensure that it has a comparable receptive field to our two-stage model. The results are shown in Table 1.

We can see that, for sequence with large shift (*Clip 020*), ours(w/ DRA) can improve the JDD-B results over ours(w/ RA) by $\sim$ 0.47dB, but it is still lags behind our full model by $\sim$1.01dB. Compared with one-stage deep alignment, our two-stage coarse-to-fine method divides the difficult large shift alignment problem into two relatively simple sub-problems, thus reduces the learning space and complexity.

DPBM vs. BM. We further evaluate the models using normal block matching (BM) and the proposed differentiable progressive block matching (DPBM) in the coarse alignment module with different search regions. The results on a clip in REDS4 are shown in Fig. 3. One can see that with the increase of search region, the performance of both models increases. However, with BM the performance reaches the peak when the search region reaches 64, while with DPBM, the performance reaches the peak when the search region is 48. This implies that DPBM can save much the computational cost. In our experiments, we set the search region as 48 for the REDS4 dataset and 128 for the Videezy4K dataset since the side length of 4K video is about three times that of 720p video.

Impact of $\mathcal{L}_{ip}$. To validate the effectiveness of our proposed $\mathcal{L}_{ip}$, we compare the models with/without $\mathcal{L}_{ip}$. The results are shown in Table 1. The model trained with $\mathcal{L}_{ip}$ (ours) achieves better PSNR/SSIM results than ours(w/o $\mathcal{L}_{ip}$). A visual comparison is shown in Fig. 4. We see that the model with $\mathcal{L}_{ip}$ achieves better structure preservation. $\mathcal{L}_{ip}$ can encourage the network to use more information from other frames to reconstruct the reference frame and avoid over-smoothing much the restoration results.

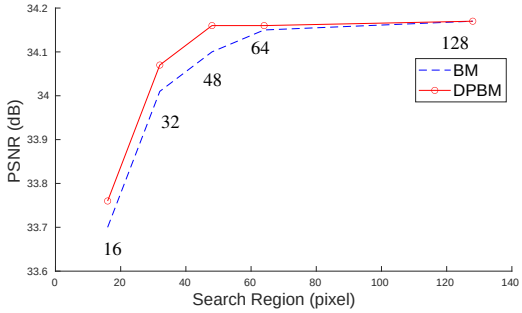


Figure 3. JDD performance of *Clip 020* in REDS4 by using BM and DPBM in the coarse alignment.

(a) Noisy image (b) Without $\mathcal{L}_{ip}$ (c) With $\mathcal{L}_{ip}$

Figure 4. Effect of $\mathcal{L}_{ip}$. The interpolation loss encourages the network to utilize more temporal information, which can improve the restoration performance on fine details.

4. More Visual Comparison Results and User Study

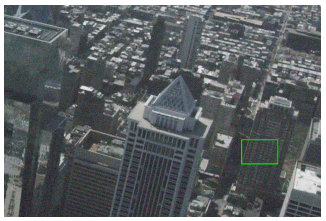
In this section, we provide more visual comparison results on images in the Videezy4K dataset and the real-world burst images in SCBurst [2]. Since there is no ground-truth in SCBurst, we further conduct a user study to evaluate the visual quality of our method.

More visual results on Videezy4K. More visual comparisons on Videezy4K are presented in Fig. 5 $\sim$ Fig. 8. In Fig. 5, we can see that the structure of the building can be well restored by using our method. Fig. 6 shows an example of 4K portrait photography. Due to the high resolution, even a small movement in the foreground can cause a large displacement between frames. One can see that our method can recover more details. In Fig. 7, one can see that our method can obtain better performance on moving objects, *i.e.* car. Fig. 8 shows the visual comparison in a small text region. Our method can restore the text more clearly, which is easier to identify.

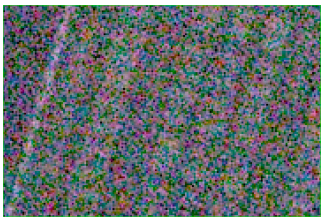
More visual results on real-world data and user study. In Figs. 9, 10 and 11, we show more visual comparison results on images in the SCBurst dataset [2], which cover moving objects, texture regions and texts. One can see that our method produces cleaner results with more fine textures.

To more comprehensively evaluate the performance of our method, we arranged a user study. We randomly cropped 120 image patches from the 16 image sequences in the SCBurst dataset, and invited 18 participants to give preference on the denoising results of different methods on the 120 patches. For images with small level noise, we randomly cropped patches in regions containing high frequency structures. For images with severe noise, the patches were randomly cropped from both high frequency areas and flat regions. The scenes of the 120 image patches are shown in Fig. 12.

The participants were asked to select the best, the second best and the third best results among the seven competing JDD-B methods, *i.e.*, KPN [3]+DMN [1], EDVR [5]+DMN [1], RviDeNet [6]+DMN [1], EDVR* [5], RviDeNet* [6], GCP-Net [2]. The 18 participants performed this user study using multiple monitors, *e.g.*, LG Ultra HD 4K display, Lecoo 4K display, Redmi 1A 1080p display, retina QHD display of Macbook Pro and LCD HD+ screen of Thinkpad X1 Carbon. The denoising results of competing methods were displayed on the screen simultaneously in random order. The interface of the user study is shown in Fig. 13. In Fig. 14, we show the statistics of the user study. In terms of the top-2 best results, one can see that our method achieves the similar number of votes to GCP-Net, both of which obtain $>$50% votes. However, our method obtains more votes than GCP-Net in terms of the top-1 results, *i.e.*, 36.1% vs. 27.9%. The user study reiterates the effectiveness of our method on JDD-B.

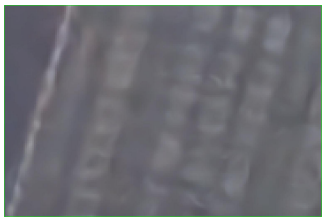


(a) Noise image



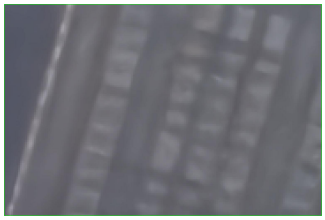
(b) Noise patch

(c) EDVR* [5]

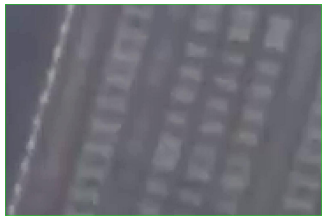


(d) RviDeNet* [6]

(e) GCP-Net [2]

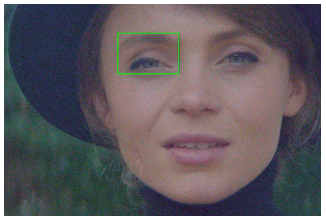


(f) Ours

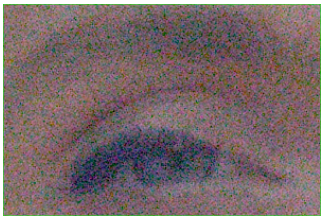


(g) GT

Figure 5. JDD results by different methods on Videezy4K dataset.

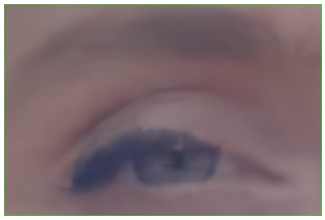


(a) Noise image

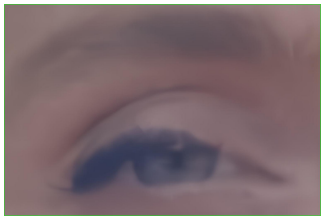


(b) Noise patch

(c) EDVR* [5]



(d) RviDeNet* [6]

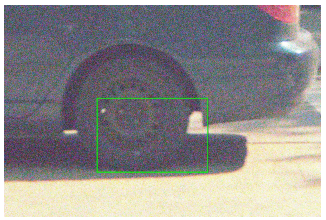


(e) GCP-Net [2]

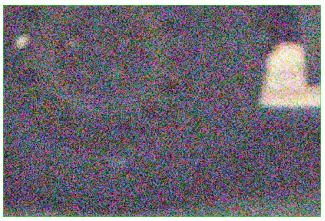
(f) Ours

(g) GT

Figure 6. JDD results by different methods on Videezy4K dataset.



(a) Noise image



(b) Noise patch

(c) EDVR* [5]

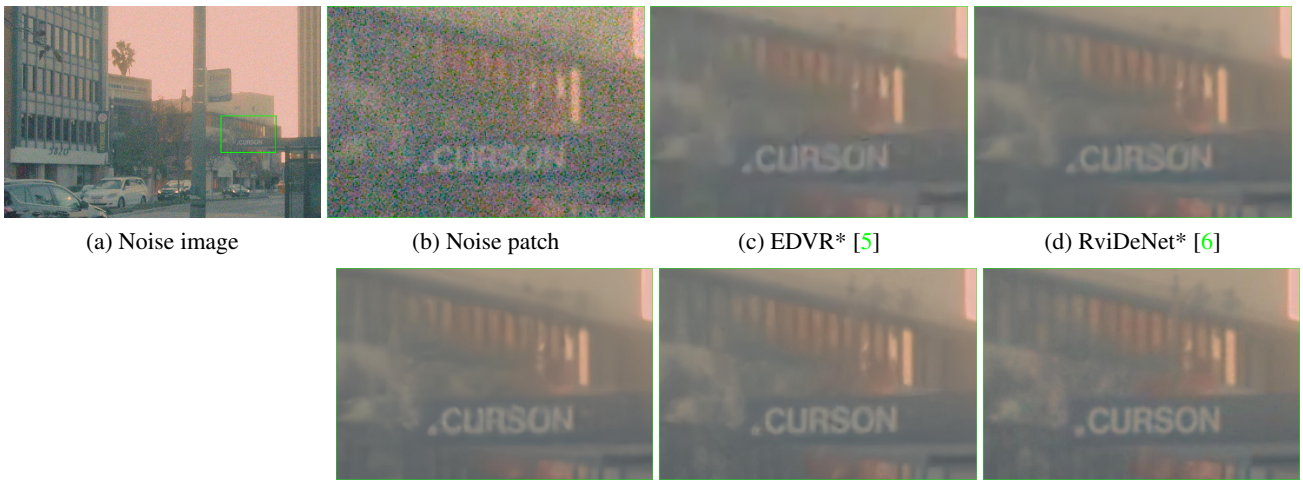
(d) RviDeNet* [6]

(e) GCP-Net [2]

(f) Ours

(g) GT

Figure 7. JDD results by different methods on Videezy4K dataset.



(a) Noise image (b) Noise patch (c) EDVR* [5] (d) RviDeNet* [6]

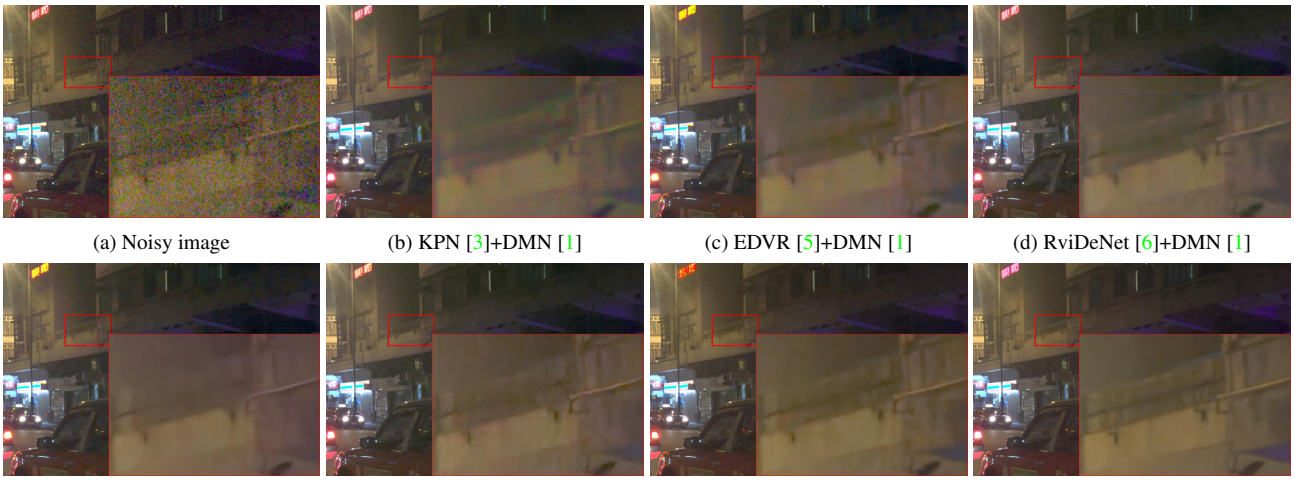
(e) GCP-Net [2] (f) Ours (g) GT

Figure 8. JDD results by different methods on Videezy4K dataset.

(a) Noisy image (b) KPN [3]+DMN [1] (c) EDVR [5]+DMN [1] (d) RviDeNet [6]+DMN [1]

(e) EDVR* [5] (f) RviDeNet* [6] (g) GCP-Net [2] (h) Ours

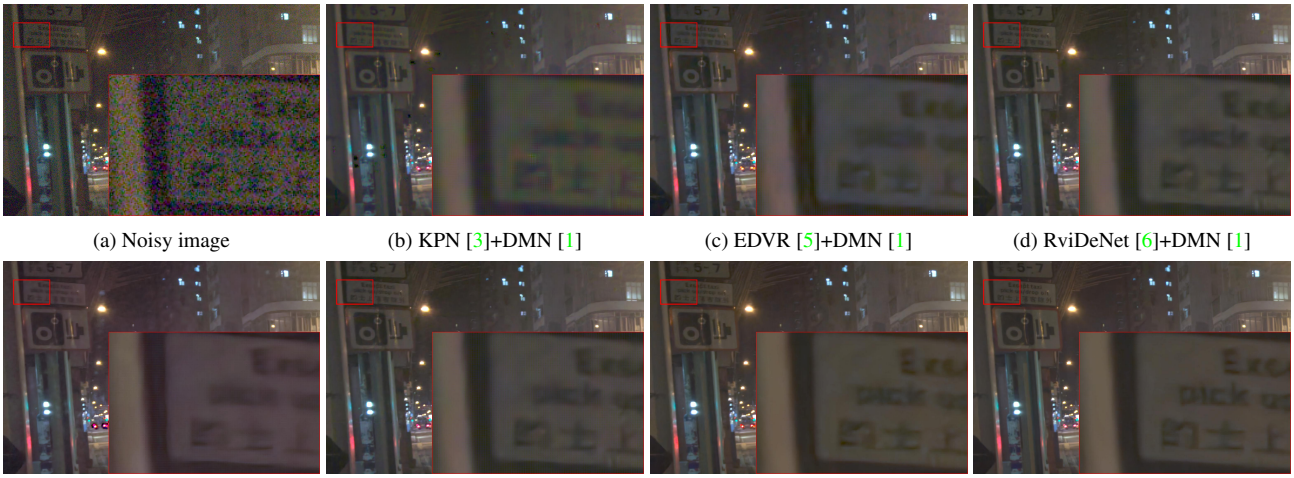
Figure 9. JDD-B results on real-world burst images by different methods.



(a) Noisy image (b) KPN [3]+DMN [1] (c) EDVR [5]+DMN [1] (d) RviDeNet [6]+DMN [1]

(e) EDVR* [5] (f) RviDeNet* [6] (g) GCP-Net [2] (h) Ours

Figure 10. JDD-B results on real-world burst images by different methods.



(a) Noisy image (b) KPN [3]+DMN [1] (c) EDVR [5]+DMN [1] (d) RviDeNet [6]+DMN [1]

(e) EDVR* [5] (f) RviDeNet* [6] (g) GCP-Net [2] (h) Ours

Figure 11. JDD-B results on real-world burst images by different methods.

Figure 12. The 120 patches used in the user study.

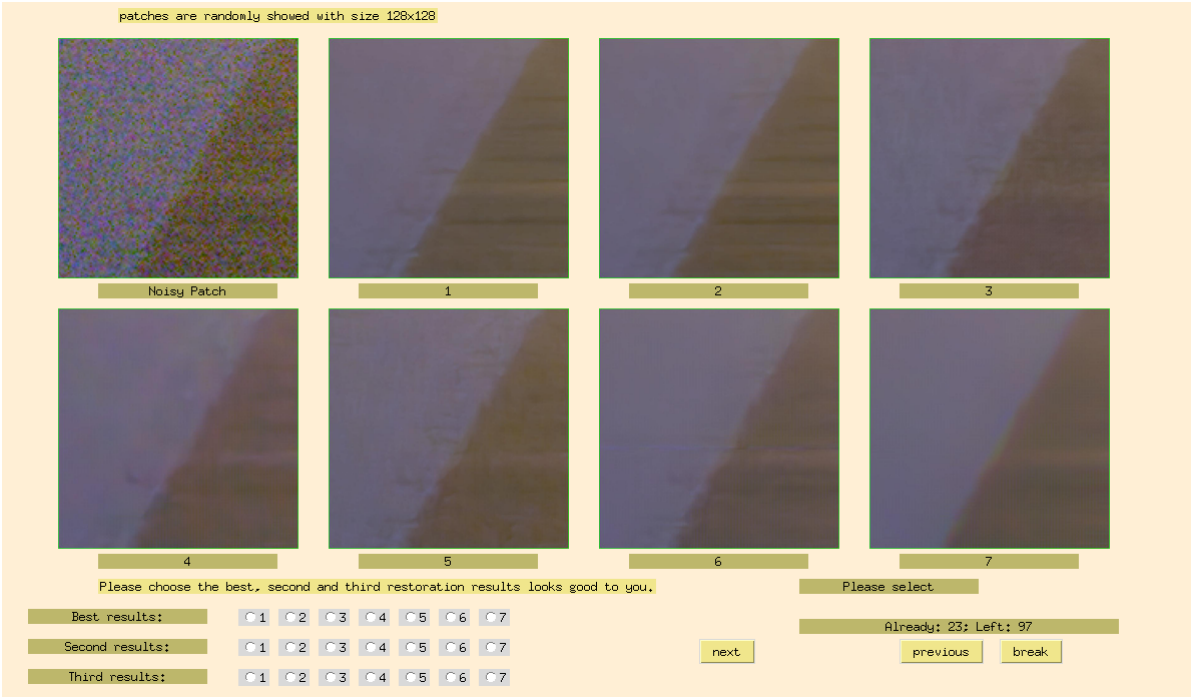


Figure 13. The interface designed for the user study.

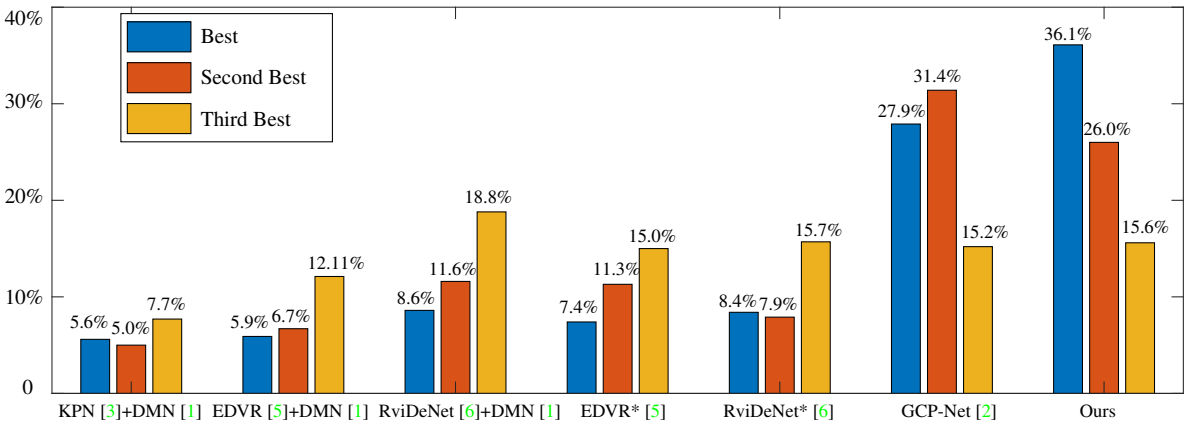


Figure 14. User study results.

References

[1] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédo Durand. Deep joint demosaicking and denoising. *ACM Transactions on Graphics (TOG)*, 35(6):1–12, 2016. 3, 5, 6, 7

[2] Shi Guo, Zhetong Liang, and Lei Zhang. Joint denoising and demosaicking with green channel prior for real-world burst images. *arXiv preprint arXiv:2101.09870*, 2021. 3, 4, 5, 6, 7

[3] Ben Mildenhall, Jonathan T Barron, Jiawen Chen, Dillon Sharlet, Ren Ng, and Robert Carroll. Burst denoising with kernel prediction networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2502–2510, 2018. 3, 5, 6, 7

[4] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pages 402–419. Springer, 2020. 1

[5] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3, 4, 5, 6, 7

[6] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2301–2310, 2020. 3, 4, 5, 6, 7