Supplementary Material: Training Weakly Supervised Video Frame Interpolation with Events

Zhiyang Yu[†1,2], Yu Zhang[*2,3], Deyuan Liu[†2,5], Dongqing Zou[2,4],
Xijun Chen[*1], Yebin Liu[3], and Jimmy Ren[2,4]

[1]Harbin Institute of Technology, [2]SenseTime Research and Tetras.AI, [3]Tsinghua University
[4]Qing Yuan Research Institute, Shanghai Jiao Tong University, [5]Peking University

1. Introduction

In this supplementary material, we elaborate the details on the following aspects:

1. **Derivations of quadratic fitting (Sect. 2)**, providing detailed mathematical derivations of the solver of the quadratic surface fitting problem as defined in Eqn. (7) of the submitted paper.
2. **Reproduction experiments of state-of-the-art models (Sect. 3)**, providing the details of how we reproduce the results of existing video frame interpolation models on the GoPro dataset.
3. **Architecture details (Sect. 4)**, providing the details of the architecture of the proposed framework.
4. **More visual results (Sect. 5)**, providing more visual comparisons between our approach and the state-of-the-art models on both synthetic and real datasets.

2. Derivations of Quadratic Fitting

In this section we elaborate the detailed derivations of the solution of Eqn. (7) in the submitted paper, *i.e.* the following least square problem:

$$\min_{\mathbf{A},\mathbf{b},c} \sum_{\mathbf{u}} \mathbf{w}(\mathbf{u}) \left\| \hat{\mathbf{d}}(\mathbf{u}) - \mathbf{d}(\mathbf{u}) \right\|^2, \tag{1}$$

in which $\hat{\mathbf{d}}(\mathbf{u})$ is defined with

$$\hat{\mathbf{d}}(\mathbf{u}) = \frac{1}{2}\mathbf{u}^{\mathrm{T}}\mathbf{A}\mathbf{u} + \mathbf{b}^{\mathrm{T}}\mathbf{u} + c,\ \mathbf{u} \in \mathbb{Z}^2 \cap [-n, n]^2, \tag{2}$$

where $\mathbb{Z}$ denotes the field of integers, and $\mathbf{d}(\mathbf{u})$ provides the known mapped values of $\mathbf{u}$ at the $(2n+1)^2$ integer lattices on the domain of $\mathbf{u}$ as defined in 2. Note that we solve 5 parameters: $\mathbf{A} = \mathrm{diag}([a_1, a_2]^{\mathrm{T}})$, $\mathbf{b} = [b_1, b_2]^{\mathrm{T}}$, and c. For $n \geq 1$, there are more than 9 known mappings in (1), making it an over-determined problem.

Denote $\mathbf{x} = [a_1, a_2, b_1, b_2, c]^{\mathrm{T}}$ as a column vector that aggregates all unknown parameters. Eqn. (1) could be reformulated with the following matrix form

$$\min_{\mathbf{x}} \left(\mathbf{y} - \mathbf{P}\mathbf{x}\right)^{\mathrm{T}} \mathbf{W} \left(\mathbf{y} - \mathbf{P}\mathbf{x}\right), \tag{3}$$

where $\mathbf{y}$ is a $(2n+1)^2 \times 1$ column vector that vectorizes $\mathbf{d}(\mathbf{u})$ and $\mathbf{P}$ is a $(2n+1)^2 \times 5$ coefficient matrix, *i.e.* the polynomial expansions of $\mathbf{x}$. The weight matrix $\mathbf{W}$ is a $(2n+1)^2 \times (2n+1)^2$ diagonal matrix that vectorizes $\mathbf{w}(\mathbf{u})$ and organizes the values on the diagonal.

Take $n = 1$ as instance. In this case, $\mathbf{y}$ takes the form

$$\mathbf{y} = \begin{bmatrix} \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(-1,-1)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(0,-1)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(1,-1)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(-1,0)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(0,0)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(1,0)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(-1,1)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(0,1)} \\ \mathbf{d}(\mathbf{u})|_{\mathbf{u}=(1,1)} \end{bmatrix}, \tag{4}$$

and $\mathbf{C}$ is instantiated with the following 9×5 matrix:

$$\mathbf{P} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & -1 & -1 & 1 \\ 0 & \frac{1}{2} & 0 & -1 & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 & -1 & 1 \\ \frac{1}{2} & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{1}{2} & 0 & 1 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & -1 & 1 & 1 \\ 0 & \frac{1}{2} & 0 & 1 & 1 \\ \frac{1}{2} & \frac{1}{2} & 1 & 1 & 1 \end{bmatrix}. \tag{5}$$

[†] The work is done during an internship at SenseTime Research.

[*] Corresponding authors: Yu Zhang (zhangyulb@gmail.com) and Xijun Chen (chenxijun@hit.edu.cn).

The diagonal weight matrix writes with

$$\mathbf{W} = \mathrm{diag}\left(\mathrm{vec}(\mathbf{w})\right), \tag{6}$$

in which $\mathbf{w}$ is a column vector organized with the same order w.r.t. $\mathbf{u}$ with that of $\mathbf{y}$.

Weighted least squares problems have closed-form solution. For Eqn. (1), the solution is calculated by

$$\mathbf{x} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{y}. \tag{7}$$

Let $\mathbf{C} = \left(\mathbf{P}^{\mathrm{T}}\mathbf{W}\mathbf{P}\right)^{-1}\mathbf{P}^{\mathrm{T}}\mathbf{W}$. Since that $\mathbf{P}$ and $\mathbf{W}$ are all constant matrices regardless of $\mathbf{y}$, computing the ith element of $\mathbf{x}$ could be achieved with a simple filtering process, through multiplying the i th row of $\mathbf{C}$ with $\mathbf{y}$. Further note that $\mathbf{y}$ is actually a vectorized representation of a local $(2n+1)\times(2n+1)$ field. Therefore, solving the fitting polynomials at each pixel position could be efficiently implemented through $(2n+1)\times(2n+1)$ convlutions (or cross-correlations) between the distance map and the filters stored in $\mathbf{C}$, which could be precomputed once.

Since that the least squares (1) is unconstrained, The estimated $\mathbf{A} = \mathrm{diag}([a_1, a_2])^{\mathrm{T}}$ may have negative diagonal parameters, up to the shape of $\mathbf{d}(\mathbf{u})$, making it not positive-definite. We address this issue by simply assigning

$$a_1 = \max(a_1, \epsilon),\ a_2 = \max(a_2, \epsilon), \tag{8}$$

where ϵ is a small constant.

3. Reproduction Experiments

Evaluation polices used by previous works on the GoPro dataset are inconsistent in several aspects. For example, SloMo [3] and QVI [10] are evaluated with 7x interpolation, while EMD [4] and EDVI [7] 10x instead. Also, QVI [10] reports the quantitative results computed on the full test images without cropping, while FLAVR [6] reports those on cropped test images. These inconsistencies make the results of existing works not directly comparable.

In the Sect. 4.1 of our paper, we aim to provide standard benchmarking results on the GoPro and SloMo-DVS datasets with unified settings, so as to ease future research in this field. By the time of submission, our reproduction experiments involve the following works which we consider representative in the literature: SloMo [3], QVI [10], DAIN [1], FLAVR [6], BHA [8] and EDVI [7], TAMI [2] and EMD [4]. In the following, we first describe the general settings of our evaluation policy, then explain the details of the reproduction experiments related to each approach.

Evaluation policy. For fair comparisons, we unify all the evaluations on 10x interpolation unless explained. We compute PSNRs and SSIMs on the full test images without any cropping. For each test video, we first sample the 11th and 21th frames as input and test on the frames in between, then 21th and 31th frames as input, and so on. Note that we exclude the first and last several frames for evalution since some exising work (*e.g.* QVI) requires a longer temporal window (e.g. consecutive 4 frames) of input frames.

Table 1. Comparing reproduction results of SloMo, QVI, DAIN and FLAVR with those of [6], using the same setting with [6].

	DAIN	SloMo	QVI	FLAVR
ours	29.01	29.78	31.57	31.31
[6]	29.00	28.52	30.55	31.31

SloMo, QVI, DAIN and FLAVR. The original version of QVI involves training on private datasets compiled by the authors, while DAIN does not report results on multi-frame interpolation on the GoPro dataset. Recently, Kalluri *et al.* [6] provides unified evaluation of these approaches, however they evaluate 7x interpolation and test on 512×512 patches cropped from the original 1280×720 test images. These settings are not consistent with ours. To this end we first reproduce the results of these methods using the same settings of [6], then adapt these verified re-implementations to train the 10x models in our setting. Note for FLAVR we directly adopt the default training parameters released by the authors, which reproduce their results. See Table 1 for comparisons between our replementations and those of [6].

EDVI. The authors of EDVI provide us the code to create and run their models. However, the training code, configuration or pretrained model on the GoPro datset is not released. Due to the lacks of sufficient details, official models or training code, evaluation policies, on the GoPro dataset, we failed to reproducing their reported results. Instead, we tried our best to search for the optimal configurations of their approach (e.g. tried different lengths of the sampled training sequences, followed the same strategies to generate events), and report the best performance in our setting. We suspect that the gap between our reproduced performance and that reported by the authors of [7] is mainly due to the inconsistency of evaluation policy. For example, the performance of TNTT [5] (which releases pretrained models on GoPro dataset) is 32.47 in PSNR as reported in the Table 1 of [7], while is however up to 29.52 as reported in a recent work [9]. Using the pretrained model, TNTT achieves 28.13 in PSNR in our evaluation, which is in similar with that of [9] however much different with that of [7].

BHA and EMD. Since that EMD adopts the similar 10x interpolation setting with that of ours on the GoPro dataset, we directly copy the results reported by the authors. BHA is also taken there, as evaluated by the authors of EMD.

Benchmarking details on SloMo-DVS dataset. We retrain the models above and the proposed approach on the SloMo-DVS dataset using the same configuration parameters without much tuning. We test BHA using the default parameter settings released by the authors.

4. Architecture Details

Architecture details are illustrated in Fig.1, 2, 3.

5. More results

More qualitative comparisons on GoPro and Slomo-DVS are illustrated in Fig.4, 5, 6, 7, 8, 9, 10, 11. More qualitative comparisons on real data can be found in https://youtu.be/ktG5U3WKGes

References

[1] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang. Depth-aware video frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3703–3712, 2019. 2

[2] Z. Chi, R. M. Nasiri, Z. Liu, J. Lu, J. Tang, and K. N. Plataniotis. All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling. In *European Conference on Computer Vision (ECCV)*, volume 12372, pages 107–123, 2020. 2

[3] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. G. Learned-Miller, and J. Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9000–9008, 2018. 2

[4] Z. Jiang, Y. Zhang, D. Zou, S. J. Ren, J. Lv, and Y. Liu. Learning event-based motion deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3317–3326, 2020. 2

[5] M. Jin, Z. Hu, and P. Favaro. Learning to extract flawless slow motion from blurry videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8112–8121, 2019. 2

[6] T. Kalluri, D. Pathak, M. Chandraker, and D. Tran. FLAVR: flow-agnostic video representations for fast frame interpolation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[7] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and S. J. Ren. Learning event-driven video deblurring and interpolation. In *European Conference on Computer Vision (ECCV)*, volume 12353, pages 695–710, 2020. 2

[8] L. Pan, C. Scheerlinck, X. Yu, R. Hartley, M. Liu, and Y. Dai. Bringing a blurry frame alive at high frame-rate with an event camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6820–6829, 2019. 2

[9] V. Rengarajan, S. Zhao, R. Zhen, J. William Glotzbach, H. R. Sheikh, and A. C. Sankaranarayanan. Photosequencing of motion blur using short and long exposures. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2150–2159, 2020. 2

[10] X. Xu, S. Li, W. Sun, Q. Yin, and M.-H. Yang. Quadratic video interpolation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1645–1654, 2019. 2

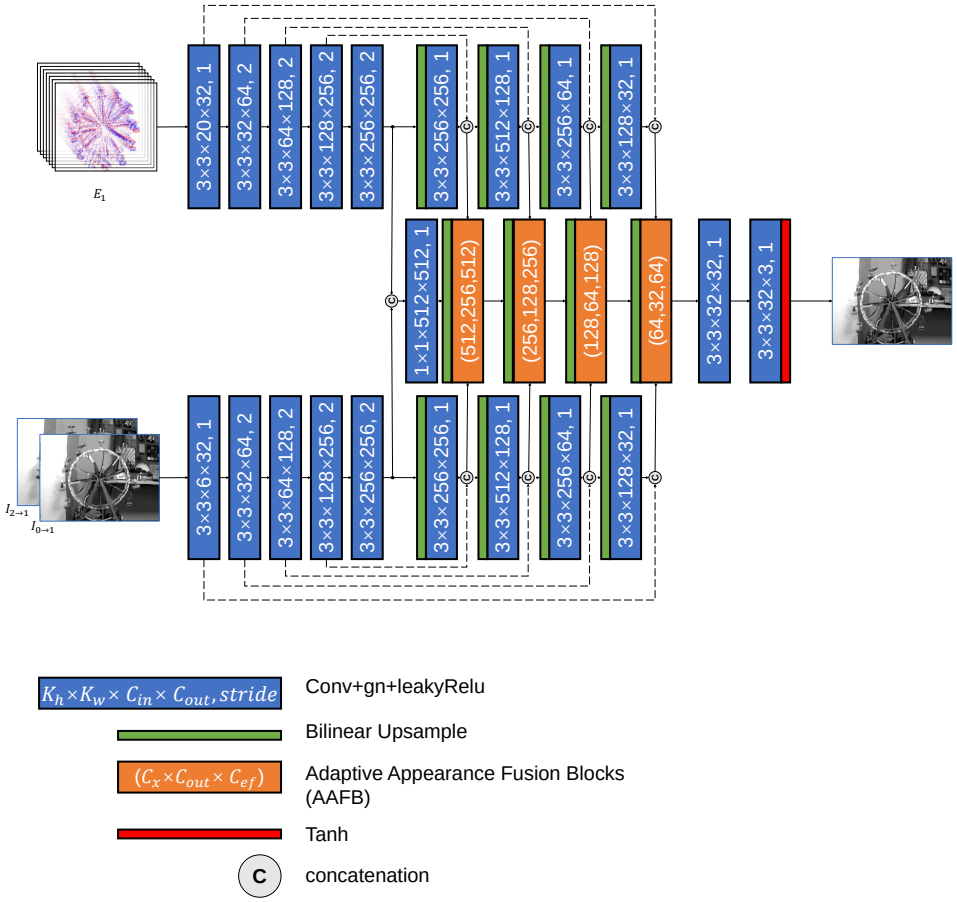

Figure 1. Architecture details of CAF, including detailed layer configurations of the two-branch Unet and multiscale branch fusion blocks. Best viewed with color.

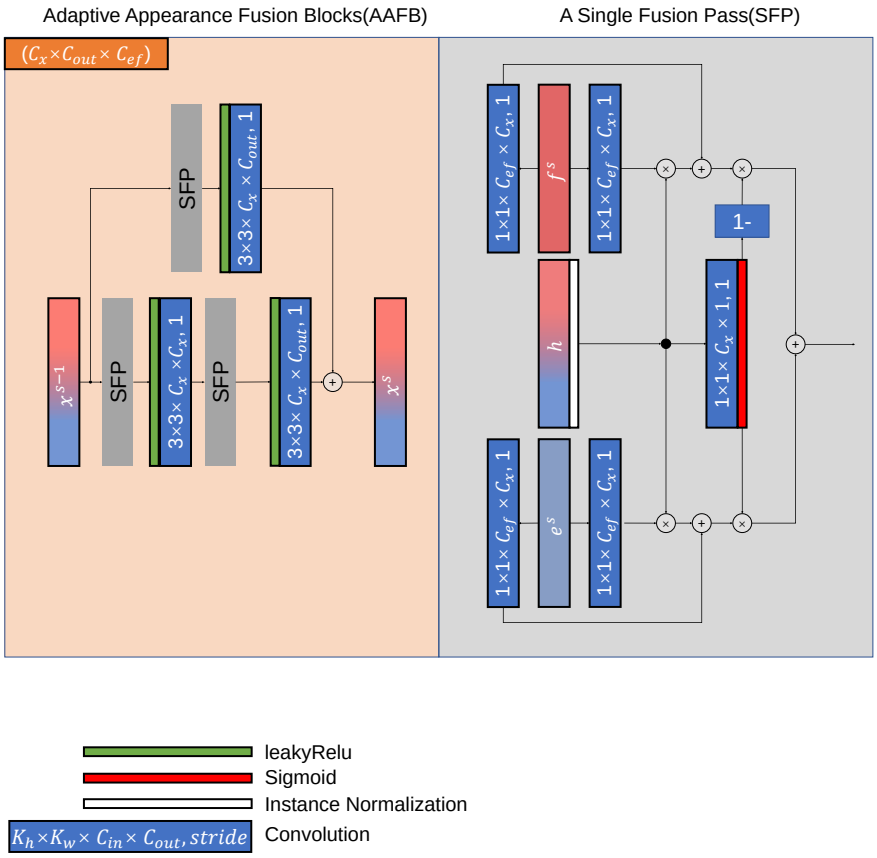

Figure 2. Architecture details of adaptive appearance fusion blocks(AAFB) and a single fusion pass(SFP). Best viewed with color.

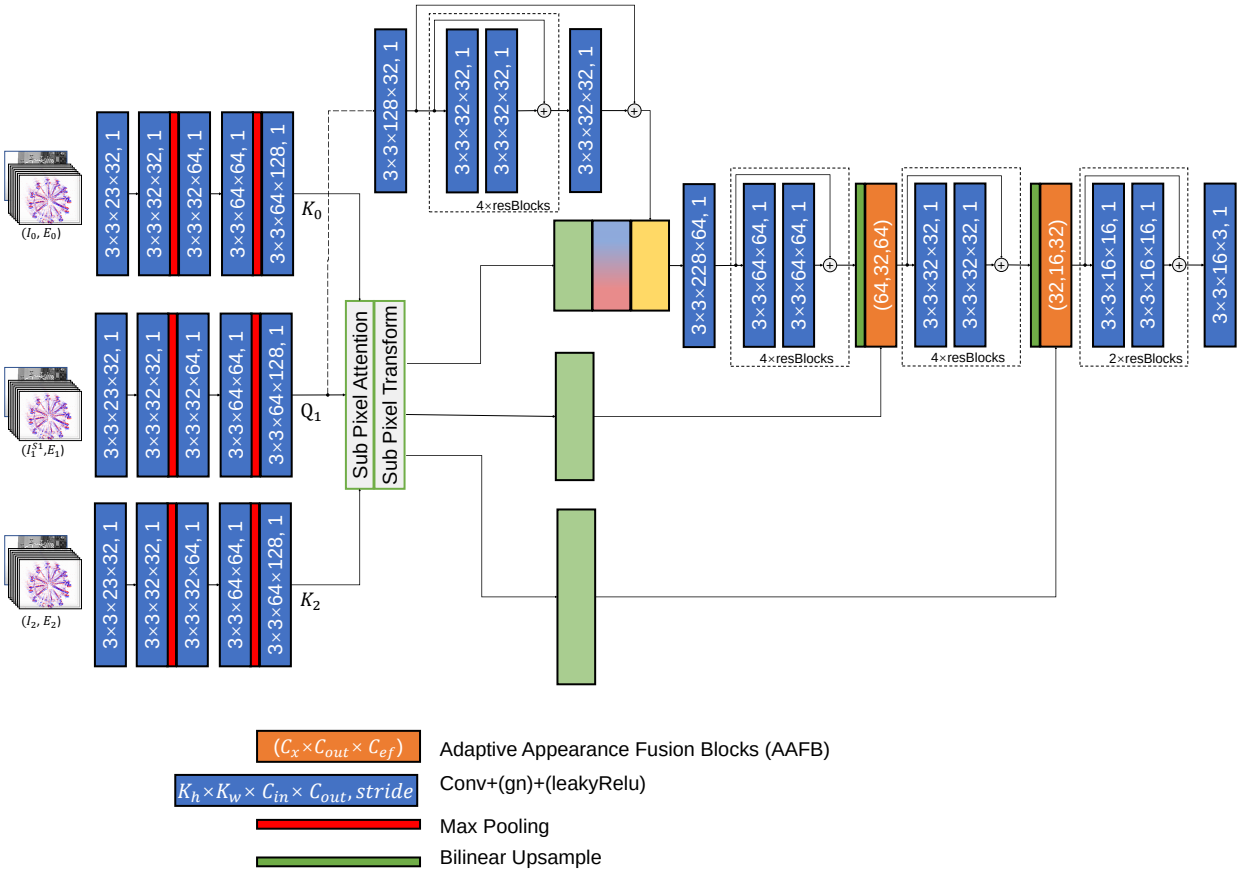

Figure 3. Architecture details of stage2. Subpixel Motion Transformer (SMT). Best viewed with color.

Figure 4. More results on GoPro. Best compared in the electronic version of this paper with zoom.

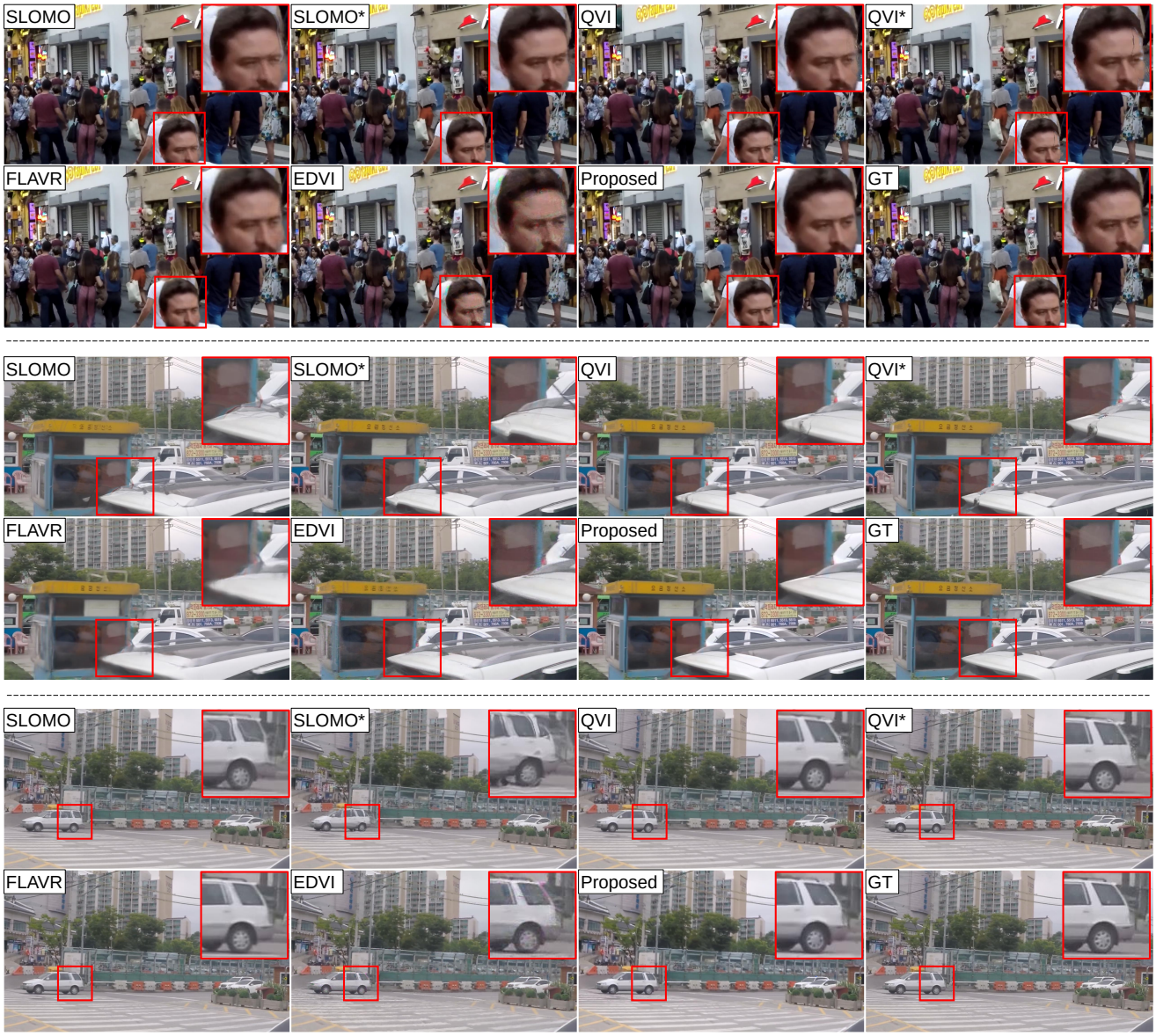

Figure 5. More results on GoPro. Best compared in the electronic version of this paper with zoom.

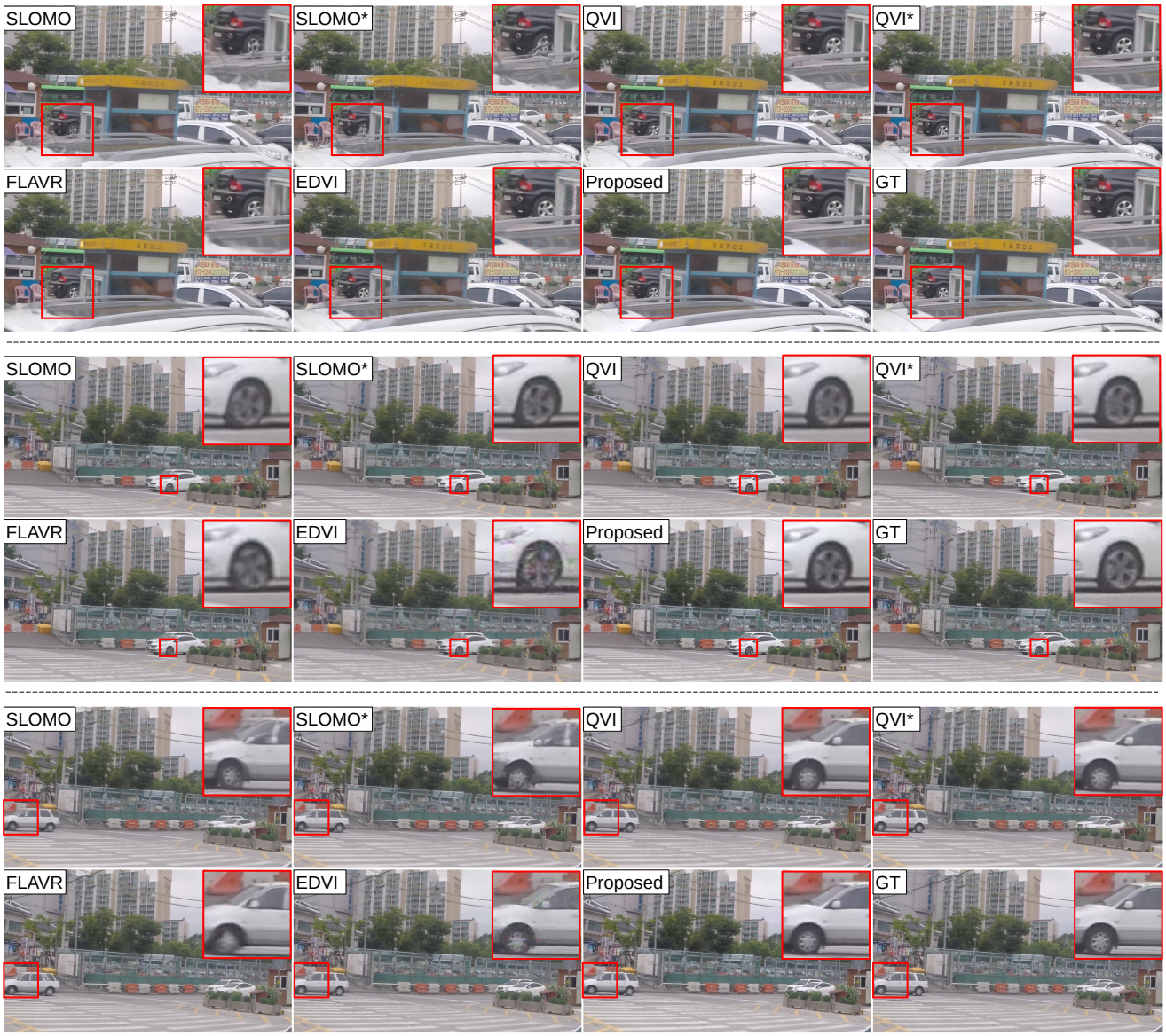

Figure 6. More results on GoPro. Best compared in the electronic version of this paper with zoom.

Figure 7. More results on GoPro. Best compared in the electronic version of this paper with zoom.

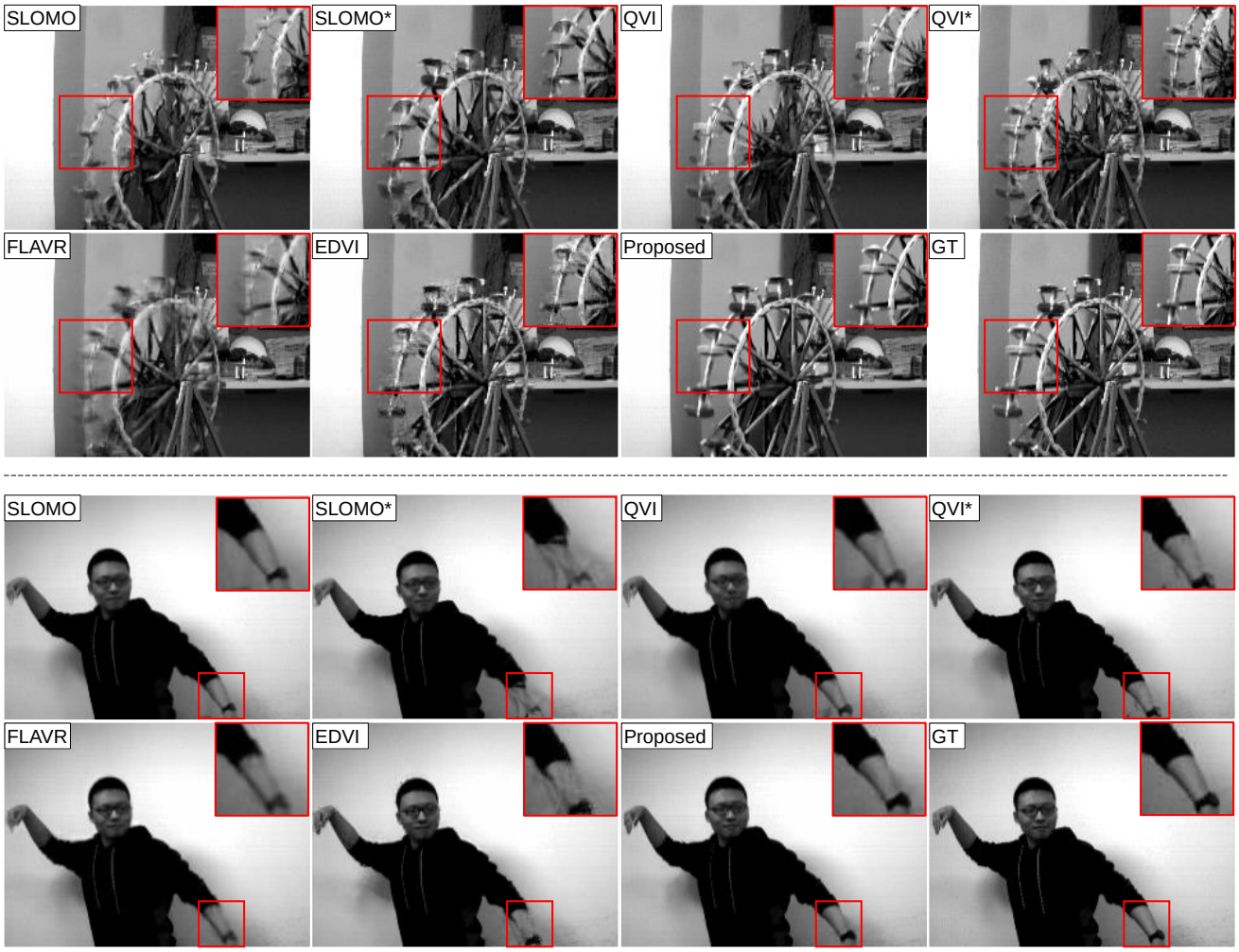

Figure 8. More results on Slomo-DVS. Best compared in the electronic version of this paper with zoom.

Figure 9. More results on Slomo-DVS. Best compared in the electronic version of this paper with zoom.

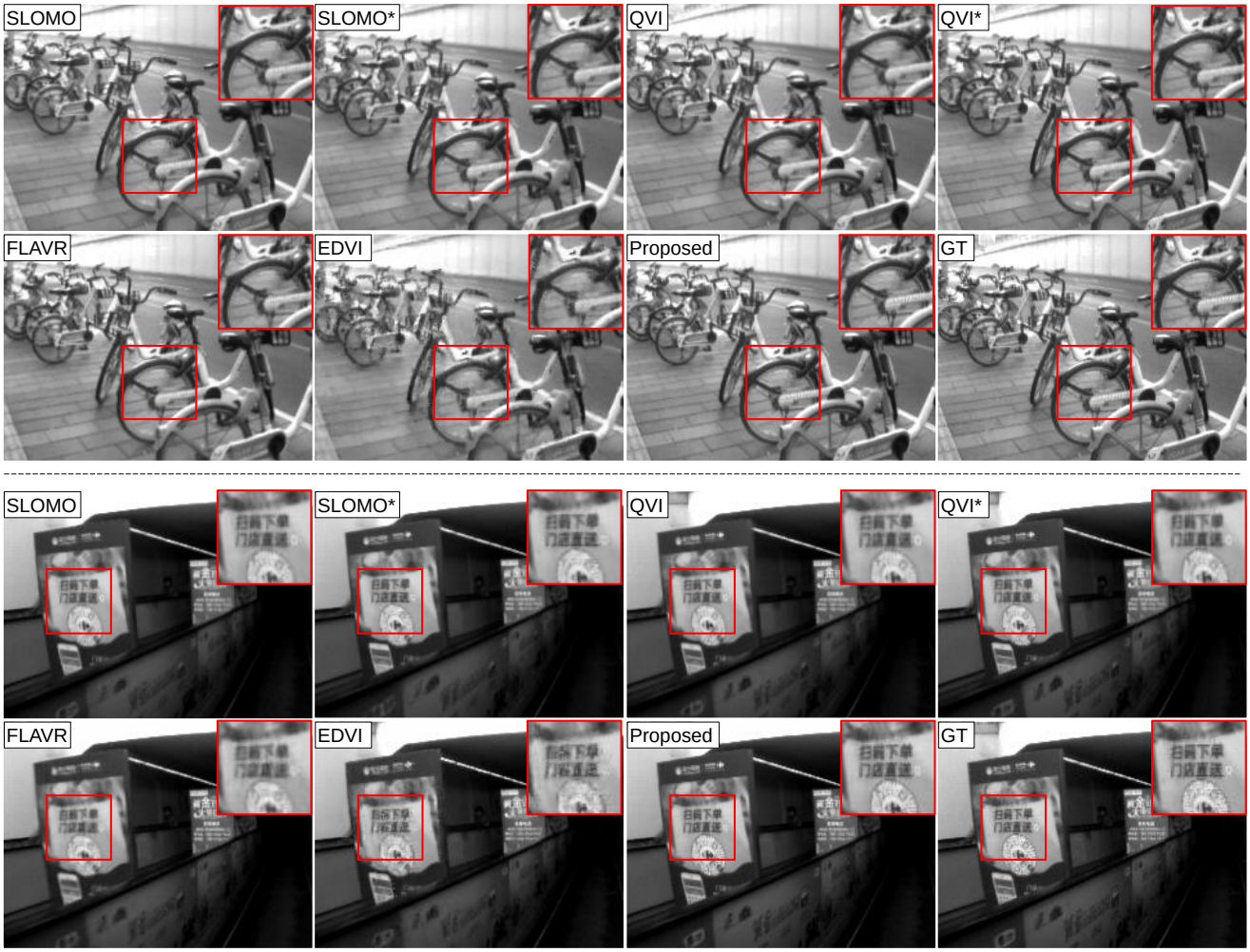

Figure 10. More results on Slomo-DVS. Best compared in the electronic version of this paper with zoom.

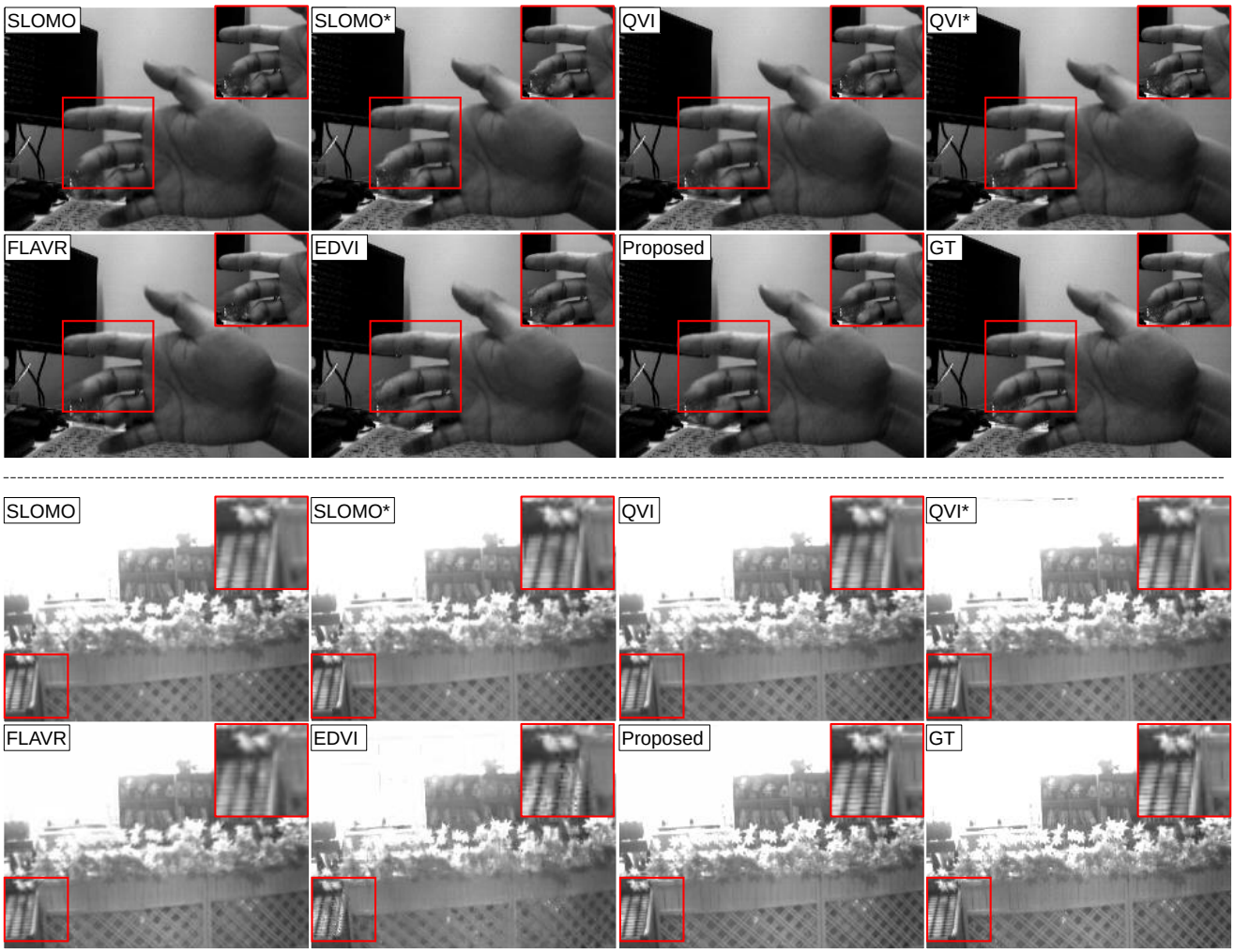

Figure 11. More results on Slomo-DVS. Best compared in the electronic version of this paper with zoom.