SensorFlow: Sensor and Image Fused Video Stabilization Supplementary Material

This supplementary material consists of four parts. In Sec. 1, we show a complete quantitative metric comparison for each category in Shi et al. [6] test set. In Sec. 2, we discuss the advantage of angular velocity based pre-stabilization and compare the camera motion with quaternion based stabilization in the existing sensor-image fusion based method Shi et al. [6]. In Sec. 3, we provide detailed network structure shown in Fig. 3 of the main paper. Finally in Sec. 4, we compare our method with industry leading video stabilization solution in iPhone 15 Pro. We encourage readers to watch our supplementary video at https://youtu.be/f8qi53KMPyY.

1. Complete Quantitative Comparison

Table 1 shows complete comparison of the stability metric. Note that since the *Driving* category only contains 2 videos and potentially introduces large variance, we merge it with the original *Parallax* category. Since our method combines the power of sensor-based stabilization and flow-based stabilization, we achieve outstanding stability for all categories. Our method especially works well with large motion videos (e.g. *Running* category, $+18.7\%$ compared to second best method), which proves the effectiveness of our angular velocity based pre-stabilization with sensor information. Methods using optical flow for motion estimation and dense warp field for stable frame generation are typically sensitive to disocclusion (e.g. *Parallax* category, $+19.6\%$ compared to second best method) and dynamic objects (e.g. *People* category, $+8.7\%$ compared to the second best method). With flow pre-processing and occlusion-aware flow stabilization network, the high performance of our method remains unaffected in the presence of parallax and moving objects.

Table 2 shows complete comparison of the distortion metric. As discussed in the main paper, our method has a comparable distortion performance with the best method PWStableNet [11], but our method is significantly more stable. Note that for the challenging cases with large motion (e.g. *Running*), our method is able to achieve more stable video ($+21.6\%$) with comparable distortion (-4.7%) to DUT [8].

Table 3 shows complete comparison of the cropping metric. Note that although our method does not focus on maintaining large field of view, the remaining region after the

Metrics	General	Rotation	Parallax	People	Running	**Average**
Input	0.4003	0.2756	0.2587	0.3453	0.2003	0.2906
Zhang et al. [10]	0.3218	0.1012	0.1721	0.2795	0.0962	0.1980
DUT [8]	0.4711	**0.3756**	0.3929	0.6587	0.4169	0.4694
Yu et al. [9]	0.4189	0.3341	0.2981	0.5339	0.3313	0.3778
PWStableNet [11]	0.4139	0.3333	0.3034	0.4132	0.2183	0.3280
Wang et al. [7]	0.3503	0.1594	0.2113	0.2501	0.2015	0.2371
Deep3D [4]	**0.5665**	0.3695	0.3547	0.5861	0.3754	0.4413
Choi et al. [2]	0.4742	0.4133	0.3296	0.4905	0.2125	0.3692
Grundmann et al. [3]	0.2739	0.2093	0.2229	0.3642	0.2013	0.2560
Shi et al. [6]	0.3285	0.2708	0.3866	0.5849	0.4272	0.4286
Ours	0.5409	0.3736	**0.4701**	**0.7162**	**0.5069**	**0.5398**

Table 1. Stability metric. Larger number indicates a better performance. Best entry is marked with **red**, and second best is marked with blue.

Metrics	General	Rotation	Parallax	People	Running	**Average**
Input	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Zhang et al. [10]	0.9558	0.7644	0.6383	0.7236	0.6277	0.7051
DUT [8]	0.9354	0.9283	0.7607	0.8704	**0.8987**	0.8457
Yu et al. [9]	0.9659	0.9110	0.7233	0.8299	0.7328	0.7905
PWStableNet [11]	0.9735	**0.9620**	**0.8978**	**0.9025**	0.8415	**0.8993**
Wang et al. [7]	0.8204	0.8723	0.6469	0.7512	0.5069	0.6745
Deep3D [4]	0.9609	0.8600	0.7967	0.8399	0.6685	0.8066
Choi et al. [2]	**0.9790**	0.9560	0.7064	0.8349	0.7439	0.7931
Grundmann et al. [3]	0.9088	0.9000	0.8270	0.8602	0.7886	0.8418
Shi et al. [6]	0.9574	0.9027	0.8211	0.8414	0.7414	0.8298
Ours	0.9564	0.9219	0.8282	0.8982	0.8561	0.8708

Table 2. Distortion metric. Larger number indicates a better performance. Best entry is marked with **red**, and second best is marked with blue.

Metrics	General	Rotation	Parallax	People	Running	**Average**
Input	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Zhang et al. [10]	0.9060	0.7405	0.6809	0.6709	0.6091	0.6989
DUT [8]	0.9394	0.8368	0.7431	0.8678	0.7242	0.7986
Yu et al. [9]	0.9804	**0.8789**	**0.8671**	0.8639	**0.8020**	**0.8680**
PWStableNet [11]	0.9314	0.8704	0.8423	0.8900	0.7571	0.8458
Wang et al. [7]	0.7818	0.6794	0.7196	0.7154	0.6801	0.7185
Deep3D [4]	**0.9916**	0.8787	0.8144	**0.9025**	0.7946	0.8563
Choi et al. [2]	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
Grundmann et al. [3]	0.7918	0.7807	0.7396	0.7519	0.7079	0.7454
Shi et al. [6]	0.8316	0.7364	0.7174	0.7285	0.5863	0.7081
Ours	0.9017	0.8110	0.8501	0.8497	0.7458	0.8332

Table 3. Cropping metric. Larger number indicates a better performance. Best entry is marked with **red**, and second best is marked with blue.

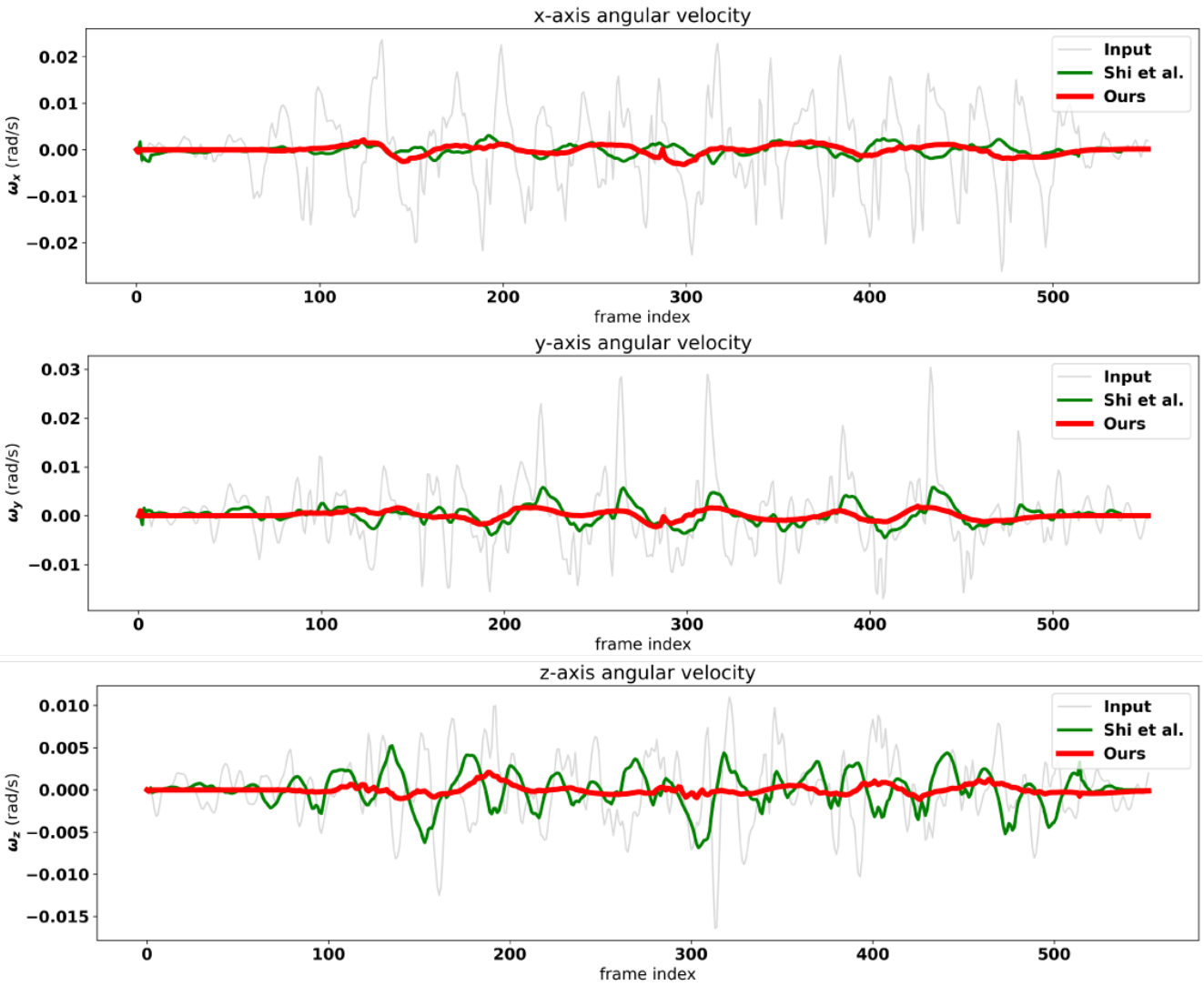

Figure 1. Camera rotation comparison between our angular velocity based pre-stabilization and the quaternion based method Shi et al. [6].

stabilization cropping is still reasonable. Compared to the best method Yu et al. [9], our cropping metric is comparable (-4.0%) and this regression is usually unnoticeable in human eye, as shown in the user study.

Other methods with sensor-based pre-stabilization. With our sensor-based pre-stabilization, other methods may achieve better results in the static scenes. However, as shown in Fig. 5 in the main paper, the most challenging part of video stabilization methods is handling dynamic scenes. Therefore, even with sensor-based pre-stabilization, other method will not achieve comparable results to our method. To support this claim, we select DUT [8] (the most stable method other than ours) and use it to stabilize the Shi et al. [6] dataset processed with our pre-stabilization. This setup achieves 0.4756 in stability, 0.7909 in distortion and 0.6781 in cropping. Note that the stability is improved slightly as expected, but the distortion is still worse due to failure dynamic object handling, and overall it is -11.9%, -9.2% and -18.6% inferior than our complete pipeline result.

2. Ablation on Angular Velocity based Pre-stabilization

Fig. 1 shows comparison of the camera motion between our angular velocity based pre-stabilization and the quaternion based optimization in Shi et al. [6]. In this figure, x-axis is the frame index and y-axis is the angular velocity along each 3D direction in radians per second; our angular velocity curve is shown in red, and Shi et al. [6] is shown in green. Note that the smaller the magnitude of angular velocity is, the more stable the video is; the smoother the angular

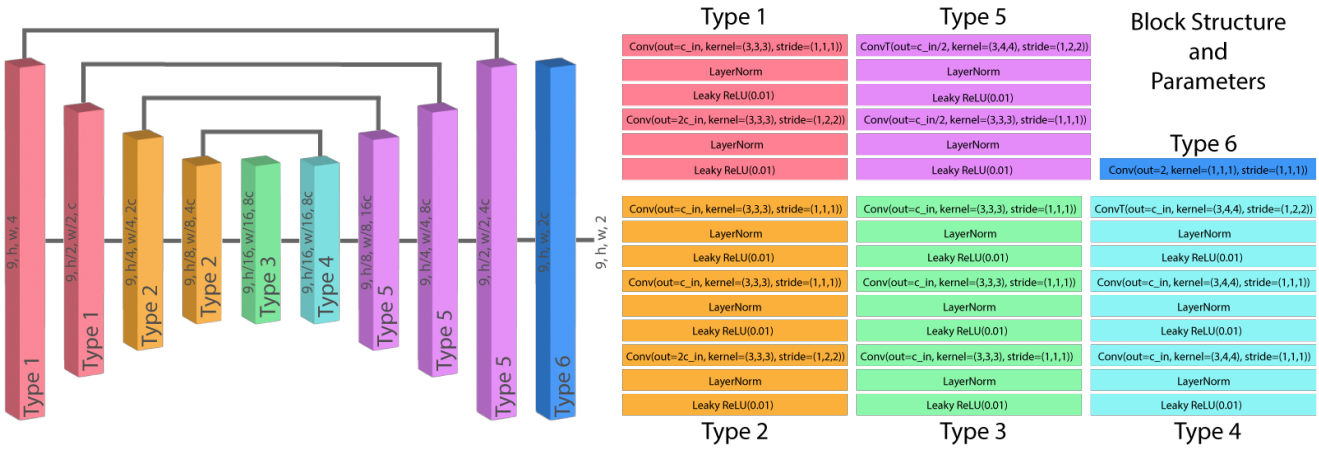

Figure 2. Network structure details and parameters.

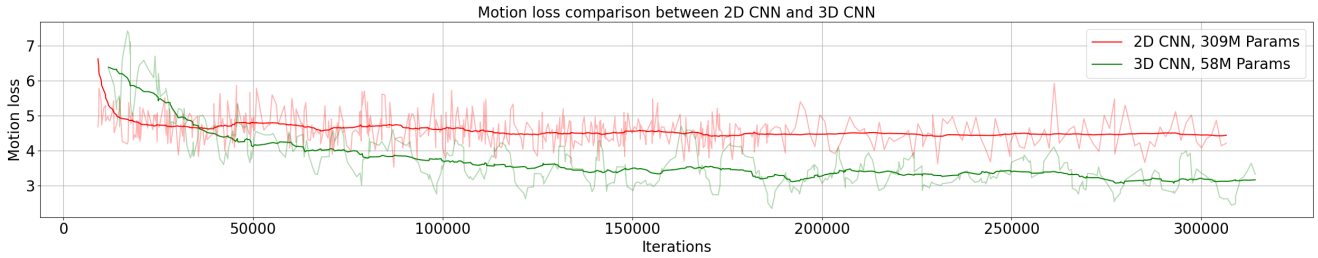

Figure 3. Motion loss comparison between 2D CNN and 3D CNN.

velocity curve is, the less jitters remains in the result video. Our method constraints smoothness directly in the velocity domain, and thus is fps invariant. The quaternion based smoothness in Shi et al. [6] assumes a fixed fps, which is not always true in practice due to sensor fps scheduling and exposure change in the input. Therefore, in Fig. 1 we can observe that our angular velocity based pre-stabilization results in both smaller angular velocity and acceleration. Our camera path is smoother in general and less affected by the frame gaps.

3. Network Structure Details

In Fig. 2, we show the detailed network structures of our flow stabilization network. We categorize the 3D convolutional blocks into 6 types, each is marked with different color. On the right of Fig. 2, we list the structure and detailed parameters used in our experiment. Note that the first *Type 1* block takes the stacked optical flow fields (with size $9 \times h \times w \times 4$) as its input, but instead of using 4 as c_{in}, we use $c_{in} = c/2$ for this block. We observe that the performance of the flow stabilization network is not sensitive to the value of c larger than 32, therefore we use $c = 32$ in all of our experiments.

Network structure discussion. We select 3D CNN as the structure of the stabilization network, instead of 2D CNN like Yu et al. [9]. The design principle is that video stabilization is a spatial-temporal process: we need to smooth the pixel tracks provided by the optical flow, while considering the rigidity of their neighborhood. To this end, 3D CNN is a more intuitive structure to directly learn temporal filtering, compared to 2D CNN that treat temporal dimension as channels and ignored the sequential information. We show the training loss comparison in Fig. 3, where 3D CNN achieves around 30% lower motion loss with around 5x less network parameters compared to 2D CNN.

4. Comparison with iPhone 15 Pro

To demonstrate the effectiveness of our method, we also compare our result and the industry leading video stabilization solution [1] in iPhone 15 Pro. To capture examples for comparison, we mount a Google Pixel 8 Pro and an iPhone 15 Pro on the same tripod side-by-side, where the former one is used to record sensor/image data for our method. Fig. 4 shows our results and the iPhone's results in two example scenes. In this figure, we consider the video sequence as a spatial-temporal volume. The images shown on the right of each example are the vertical center slice of this volume similar to the visualization in Liu et al. [5]. Note

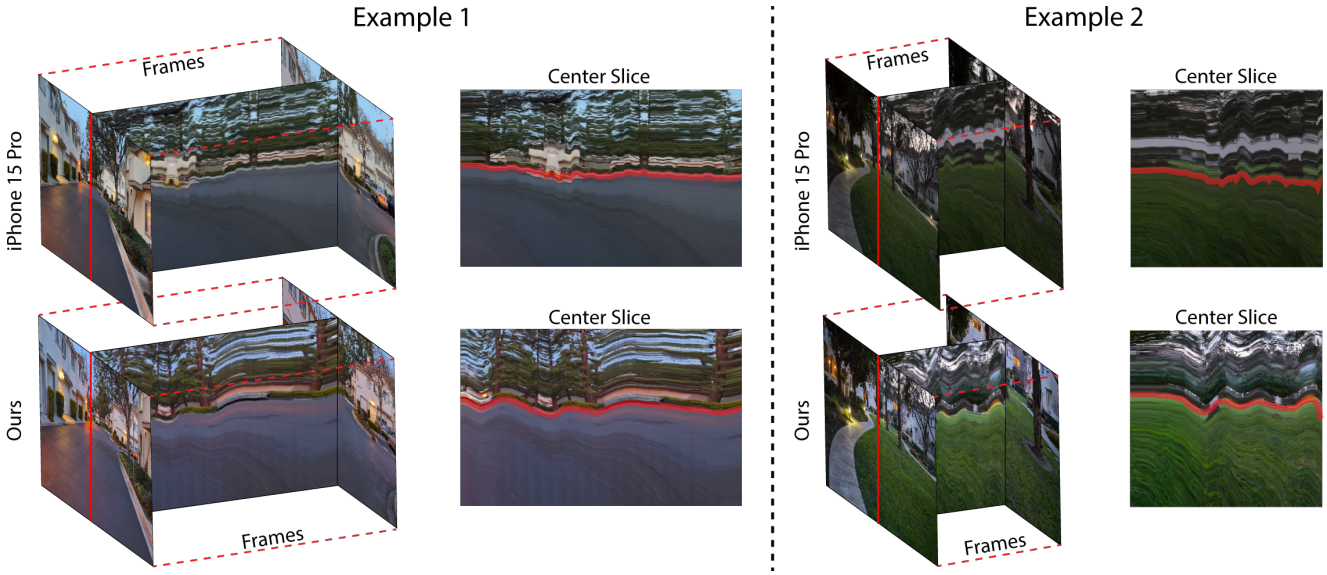

Figure 4. Comparison with industry leading video stabilization solution in iPhone 15 Pro. Vertical center slice are shown on the right of each example. Red curve marks the trajectory of a small region in the video. Our result has significantly less jitters due to learning based flow stabilization.

that the shakes in the video will be reflected by local deformation in the slice. In Fig. 4, we mark the trajectory of a small region with red curve. It can be observed that the curve for iPhone has many local deformations. Each deformation represents a jitter in the video. Our result achieves more stable results with less jitters compared to iPhone 15 Pro's stabilization results. For better visual comparison, we encourage reader to watch our supplementary video.

References

[1] Apple iPhone 15 Pro Camera test - DXOMARK. https://www.dxomark.com/apple-iphone-15-pro-camera-test/, 2023. 3

[2] Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM TOG*, 39(1):1–9, 2020. 1

[3] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011. 1

[4] Yao-Chih Lee, Kuan-Wei Tseng, Yu-Ta Chen, Chien-Cheng Chen, Chu-Song Chen, and Yi-Ping Hung. 3d video stabilization with depth estimation by cnn-based optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10621–10630, 2021. 1

[5] Yu-Lun Liu, Wei-Sheng Lai, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Hybrid neural fusion for full-frame video stabilization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2299–2308, 2021. 3

[6] Zhenmei Shi, Fuhao Shi, Wei-Sheng Lai, Chia-Kai Liang, and Yingyu Liang. Deep online fused video stabilization. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 1250–1258, 2022. 1, 2, 3

[7] Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Song-Hai Zhang, Ariel Shamir, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE TIP*, 2018. 1

[8] Yufei Xu, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. Dut: Learning video stabilization by simply watching unstable videos. *IEEE TIP*, 2022. 1, 2

[9] Jiyang Yu and Ravi Ramamoorthi. Learning video stabilization using optical flow. In *CVPR*, 2020. 1, 2, 3

[10] Zhuofan Zhang, Zhen Liu, Ping Tan, Bing Zeng, and Shuaicheng Liu. Minimum latency deep online video stabilization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23030–23039, 2023. 1

[11] Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE TIP*, 2020. 1