Deep Online Fused Video Stabilization - Supplementary Materials

Zhenmei Shi
University of Wisconsin Madison

Fuhao Shi
Google

Wei-Sheng Lai
Google

Chia-Kai Liang
Google

Yingyu Liang
University of Wisconsin Madison

In this supplementary materials, we provide more details to the network architecture, a short introduction to the OIS, the definitions of our evaluation metrics, execution time comparisons, description of our dataset, and more visual comparisons.

1. Implementation Details

Network architecture. Table 1 shows the detailed network configuration of our model. Given an input video, we resize the input frames to extract forward and backward flows at the spatial resolution of 480×270. Our encoder consists of 5 strided convolutional layers that map the optical flow from $4 \times 270 \times 480$ to $128 \times 1 \times 1$. Each convolutional layer is followed by a ReLU activation function. The latent code is converted to $1 \times 1 \times 64$ by an FC layer and then concatenated with the real and virtual pose histories. Therefore, our latent motion representation is a 188-channel feature vector, where 64 channels are from the encoded optical flow, 21×4 channels are from the real poses, and 10×4 channels are from the virtual poses. The hidden layers in the LSTM have 512 channels, and the last FC layer converts the output of LSTM to a 4D quaternion as the virtual pose of the current frame. We apply a soft-shrink activation function [6] in the last FC layer to smooth the prediction.

Boundary handling. Our method takes 10 past frames and 10 future frames from the real pose history. To ensure that our model can generate full-length videos without dropping any frame, we apply temporal padding to the input video. Specifically, we reversely pad the real pose history of the first 10 frames at the beginning and the last 10 frames at the end of the video. The virtual pose history is initialized by copying the real pose history at the beginning. We note that the reverse temporal padding is critical to initialize our model and LSTM. A simple repeated or zero padding results in worse initialization and stabilization results as shown in Fig. 1.

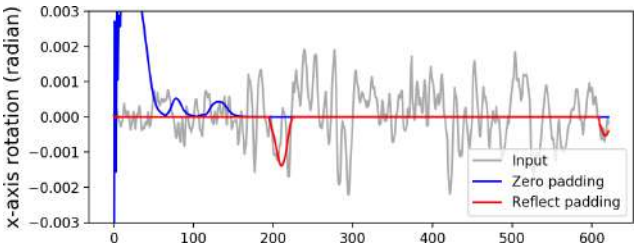

Figure 1: **Comparison on temporal padding.** With zero padding for the real and virtual camera poses, the predicted virtual poses are not stable at the beginning of a sequence.

Layer	Kernel	Strided	Input size	Output size
Conv1	3×3	1	$4 \times 270 \times 480$	$8 \times 270 \times 480$
Conv2	3×3	1	$8 \times 270 \times 480$	$16 \times 270 \times 480$
MaxPool2	4×4	4	$16 \times 270 \times 480$	$16 \times 67 \times 120$
Conv3	3×3	1	$16 \times 67 \times 120$	$32 \times 67 \times 120$
MaxPool3	4×4	4	$32 \times 67 \times 120$	$32 \times 16 \times 30$
Conv4	3×3	1	$32 \times 16 \times 30$	$64 \times 16 \times 30$
MaxPool4	4×4	4	$64 \times 16 \times 30$	$64 \times 4 \times 7$
Conv5	3×3	1	$64 \times 4 \times 7$	$128 \times 4 \times 7$
MaxPool5	4×4	4	$128 \times 4 \times 7$	$128 \times 1 \times 1$
FC1	-	-	$128 \times 1 \times 1$	64
Concat	-	-	$64 + 84^* + 40^\dagger$	188
LSTM1	-	-	188	512
LSTM2	-	-	512	512
FC2	-	-	512	4

Table 1: **Network configuration.** $*$ and † indicate the real and virtual camera poses, respectively.

2. Optical Image Stabilizer

In this section we briefly describe the functionality of optical image stabilizer (OIS) and why it is not enough for video stabilization. OIS is a mechanical approach to adjust the camera systems for compensating the camera motion and avoid motion blur in the captured images. On bulky DSLR, this is achieved by moving a few lens pieces in the compound lens [13] or the imaging sensor [8]. OIS is even more crucial on smartphones due to their limited sensor size and light-collection efficiency, and it is usually imple-

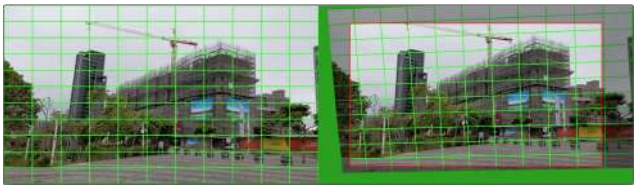

Figure 2: **Grid-based frame warping.** Left: the input frame with a regular 12x12 grid. Right: the warped frame with a virtual camera pose. The red bounding box shows the output frame after cropping. Note the rolling shutter distortion is also corrected (curved lines are straightened).

mented by moving the whole lens module [1]. Almost all premium or high-tier smartphones equip their cameras with OIS since 2016.

To detect and compensate the motion during the exposure, OIS systems typically rely on high-frequency and precise inertial measurement sensors (up to a few thousand Hz) to provide the motion signal and place the closed-loop control circuit near the moving hardware components to minimize the latency. While the real-time compensation can eliminate much motion blur, OIS has its own limitations. First, OIS usually has a limited working range (only 1-2 degrees for the smartphones) before it reaches the physical limitation (the cage housing the camera module). Moreover, when the optical elements shifts too far from the ideal position, the resulting image can suffer from other artifacts such as vignetting or inaccurate focusing. Finally, the OIS in smartphones can create some wobbling artifacts to the video because it usually uses translation to approximate the rotational model [9].

Due to these limitations, digital processing is still required to handle large motions encountered in video recording. As some parts of the camera motion is compensated by OIS, the remaining motions in the videos can not be solely represented by the gyroscope signal. Therefore, the impact of the OIS must be taken into account in the camera model [9, 11]. Our solution takes the OIS motion into account, and the dataset will come with the codes to model the motions introduced by OIS.

3. Grid-based Frame Warping

We use a grid-based warping similar to Karpenko et al. [7] to jointly stabilize video frames and remove the rolling shutter distortion. For each frame, we record the timestamp at the start of frame exposure t_f, length of rolling shutter l_{rs}, exposure duration l_{exp}, and other frame metadata (e.g., focal length, sensor size). We divide a frame into M columns and S horizontal stripes, where each stripe has its unique timestamp (see Fig. 2). By warping all stripes to a virtual camera pose P_v, rolling shutter distortion is corrected. Specifically, the warping grid is generated as

$$x_v(i,j) = K_v R_v R_r^{-1}(t_i) K_r^{-1}(t_i) x_r(i,j), \qquad (1)$$

where $t_i = t_f + l_{exp}/2 + l_{rs}/S * i$ is the stripe timestamp at row i. $x_r(i,j)$ is the 2D location on row i and column j. We set the mesh dimension to 12×12 in all the experiments.

4. Evaluation Metrics

We adopt the following metrics to evaluate the quality of stabilized videos. Note that the FOV ratio[1], stability, and distortion are commonly used metrics in prior works [2, 10, 14, 15], while the correlation is used to measure the local distortion. For all of them, a higher value indicates a better quality.

FOV ratio: We first fit a homography between the input and stabilized frames. The FOV ratio is defined as the minimal scale components of the homography across the entire video. We note that while Choi et al. [2] generate full-frame stabilization results, their FOV ratios calculated from this metric are usually smaller than 1.0 due to the homography fitting (where the average is 0.917 in our test set). As Choi et al. [2] do not apply any cropping to their results, we set their FOV ratio to 1 in Table 1 of the main manuscript for fair comparisons.

Stability: We first compute the frame-to-frame homography transformation on the stabilized video. Then, we extract the translation magnitudes and rotation angles from the homography of each frame to generate two 1D profiles. We compute the ratio between the sum of low-frequency components (2^{nd} to 6^{th}) and the total energy in the FFT domain. The stability score of a video is the average of the two ratios from translation and rotation. The intuition here is that a video is more stable if more energy is contained in the low-frequency components of the motion profile.

Distortion: We fit a homography between the input and stabilized frames and then measure the ratio of the two largest eigenvalues of the affine part from the homography. The minimal ratio from all the frames is chosen as the distortion score of a video.

Correlation: The distortion score measures global geometry distortion but cannot reflect local distortion or visual artifacts. Therefore, we measure the local distortion using a tile-based correlation. Specifically, we first align the input frame to the stabilized frame using a homography transform. To avoid any undefined regions, we experimentally choose a inner region (0.1 crop on each side) and split into a 6×6 tile on each frame. We measure the average correlation score of the entire video.

[1]This metric is named as "cropping ratio" in [2, 10, 14, 15]. However, the definition is counterintuitive, where a large ratio refers to less crop. Therefore, we rename it to "FOV ratio" in this work.

Table 2: **Execution time**. We test each method on a video with resolution 1920×1080 and average the execution time for 500 frames.

Method	CPU/GPU	Second per frame
Grundmann et al. [3]	CPU	0.541
Wang et al. [12]	GPU	0.576
PWStableNet [16]	GPU	0.108
Yu et al. [15]	GPU	2.967
Choi et al. [2]	GPU	6.505
Ours	CPU + GPU	0.287

Table 3: **Timing breakdown of our method**.

Stage	CPU/GPU	Second per frame	Percentage
Optical flow	GPU	0.134	46.7%
DNN	GPU	0.023	8.0%
Warping	CPU	0.130	45.3%

5. Execution Time

We evaluate the execution time of the proposed method and state-of-the-art approaches [2, 3, 12, 15, 16] on a machine with Intel Xeno CPU and Tesla T4 GPU. We test on a video with 1920×1080 resolution and calculate the average execution time of 500 frames in Table 2. We also provide a breakdown of our method in Table 3. Note that our core algorithm (the DNN part) is very efficient and only takes 8% of the execution time. The speed of optical flow estimation can be further improved by adopting efficient flow models [4, 5]. Our grid warping is based on a CPU implementation and can be sped-up with GPU optimization. In addition, the computational cost of optical flow estimation and grid warping can also be shared and hidden in the video processing pipelines, e.g., video recording on smartphones.

6. Video+Sensor Dataset

We collect 50 videos with Google Pixel 4 along with the gyroscope and the OIS readings. The snapshots of these videos are shown in Fig. 3 and 4 with tags on its scene and motion properties. We cover several different motions: static, panning at different speed, tracking subjects, running, circling around the subject, and riding on the vehicle. The scenes include all lighting conditions from bright daylight to low-light at night. The subjects include static objects at various depths, moving/deforming humans and high-speed vehicles.

7. Visual Comparisons and Video Results

We provide more visual comparisons with state-of-the-arts [2, 3, 12, 15, 16] in Fig. 5, 6, 7, 8, and 9. We also provide a demo video that include the full video results and the ablation study.

References

[1] Yu-Han Chen. Image stabilizer and image capturing device. US Patent 9,046,736, 2015. 2

[2] Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM TOG*, 39(1):1–9, 2020. 2, 3, 9

[3] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011. 3, 5

[4] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. Liteflownet: A lightweight convolutional neural network for optical flow estimation. In *CVPR*, 2018. 3

[5] Tak-Wai Hui, Xiaoou Tang, and Chen Change Loy. A lightweight optical flow cnn-revisiting data fidelity and regularization. *arXiv:1903.07414*, 2019. 3

[6] Kenzo Isogawa, Takashi Ida, Taichiro Shiodera, and Tomoyuki Takeguchi. Deep shrinkage convolutional neural network for adaptive noise reduction. *IEEE Signal Processing Letters*, 25(2):224–228, 2017. 1

[7] Alexandre Karpenko, David Jacobs, Jongmin Baek, and Marc Levoy. Digital video stabilization and rolling shutter correction using gyroscopes. *Stanford University Computer Science Tech Report*, 1:2, 2011. 2

[8] Sumio Kawai and Takashi Okamura. Drive device and image stabilizer. US Patent 7,973,822, 2011. 1

[9] Chia-Kai Liang and Fuhao Shih. Fused video stabilization on the Pixel 2 and Pixel 2 XL. https://ai.googleblog.com/2017/11/fused-video-stabilization-on-pixel-2.html, 2017. 2

[10] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM TOG*, 32(4):78:1–78:10, 2013. 2

[11] Damien J Thivent, George E Williams, Jianping Zhou, Richard L Baer, Rolf Toft, and Sebastien X Beysserie. Combined optical and electronic image stabilization. US Patent 9,979,889, 2018. 2

[12] 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. 3, 6

[13] Koichi Washisu. Optical image stabilizer and optical apparatus. US Patent 7,983,546, 2011. 1

[14] Jiyang Yu and Ravi Ramamoorthi. Robust video stabilization by optimization in cnn weight space. In *CVPR*, 2019. 2

[15] Jiyang Yu and Ravi Ramamoorthi. Learning video stabilization using optical flow. In *CVPR*, 2020. 2, 3, 8

[16] Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE TIP*, 2020. 3, 7

Figure 3: **Thumbnails of our training set with tags on the characteristics of each video.**

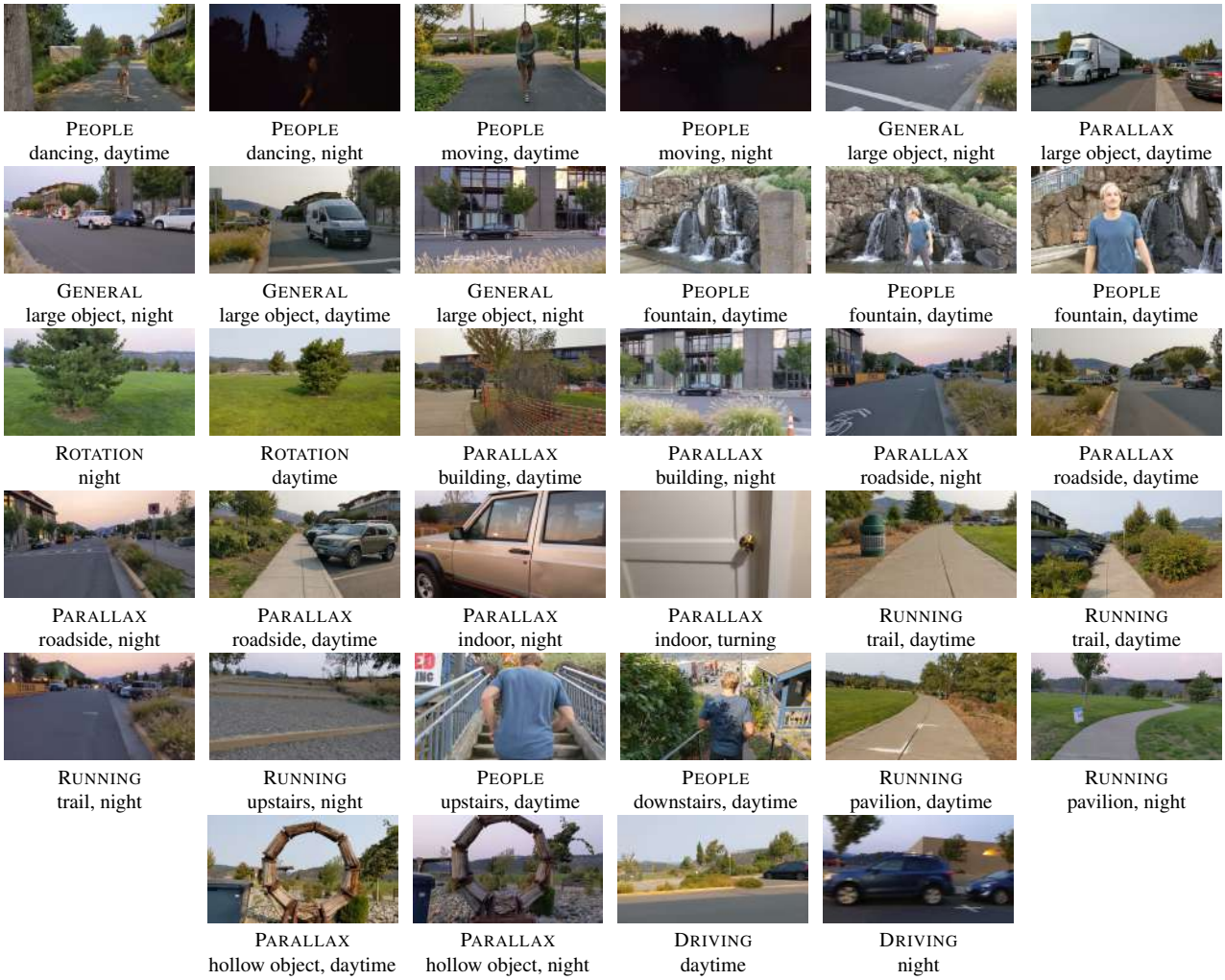

Figure 4: **Thumbnails of our test set with tags on the characteristics of each video.**

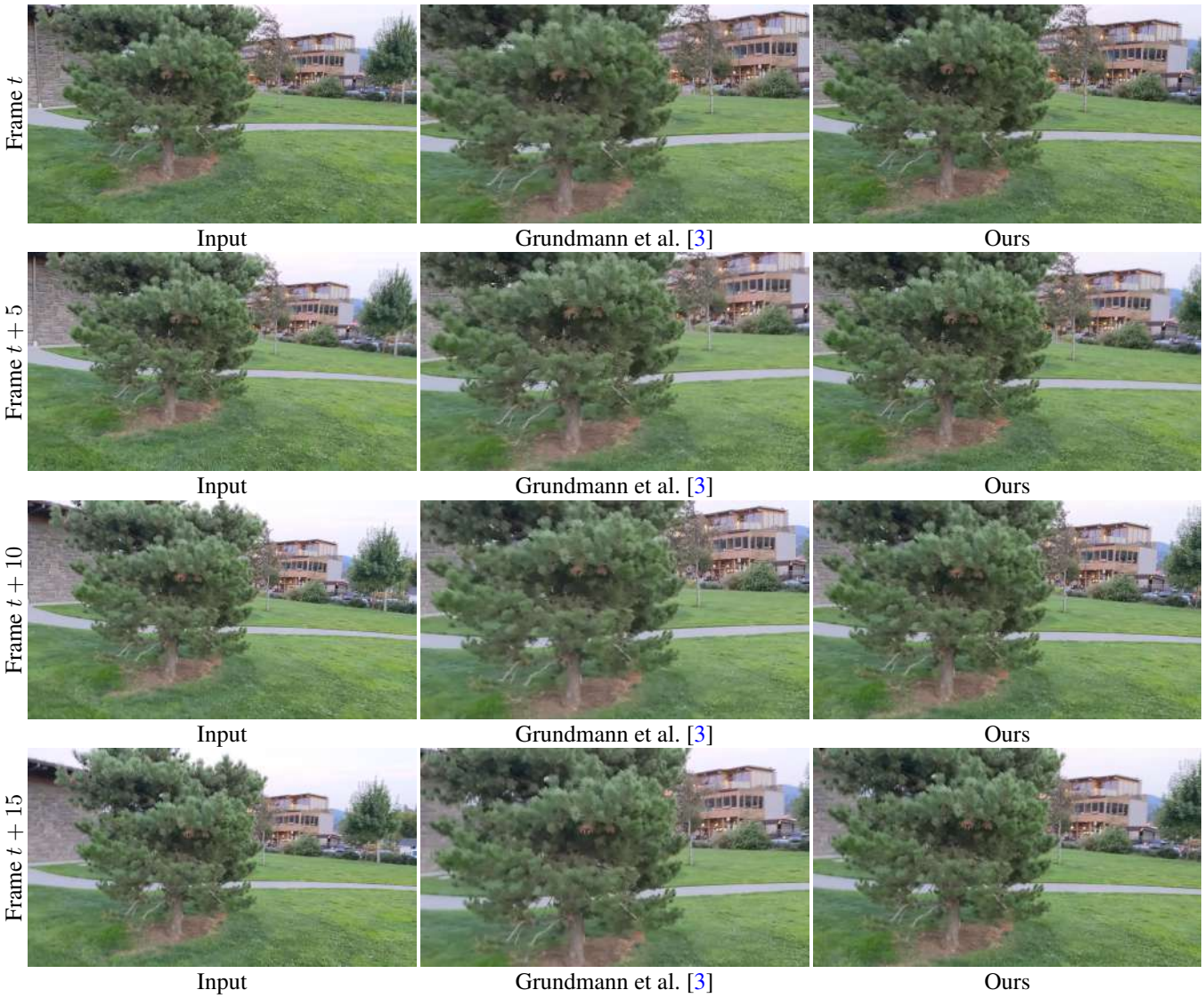

Figure 5: **Visual comparisons with Grundmann et al. [3].** The results of Grundmann et al. [3] contain residual shaking (e.g., the sky area on the top right corner is decreasing and then increasing), while our results are more stable.

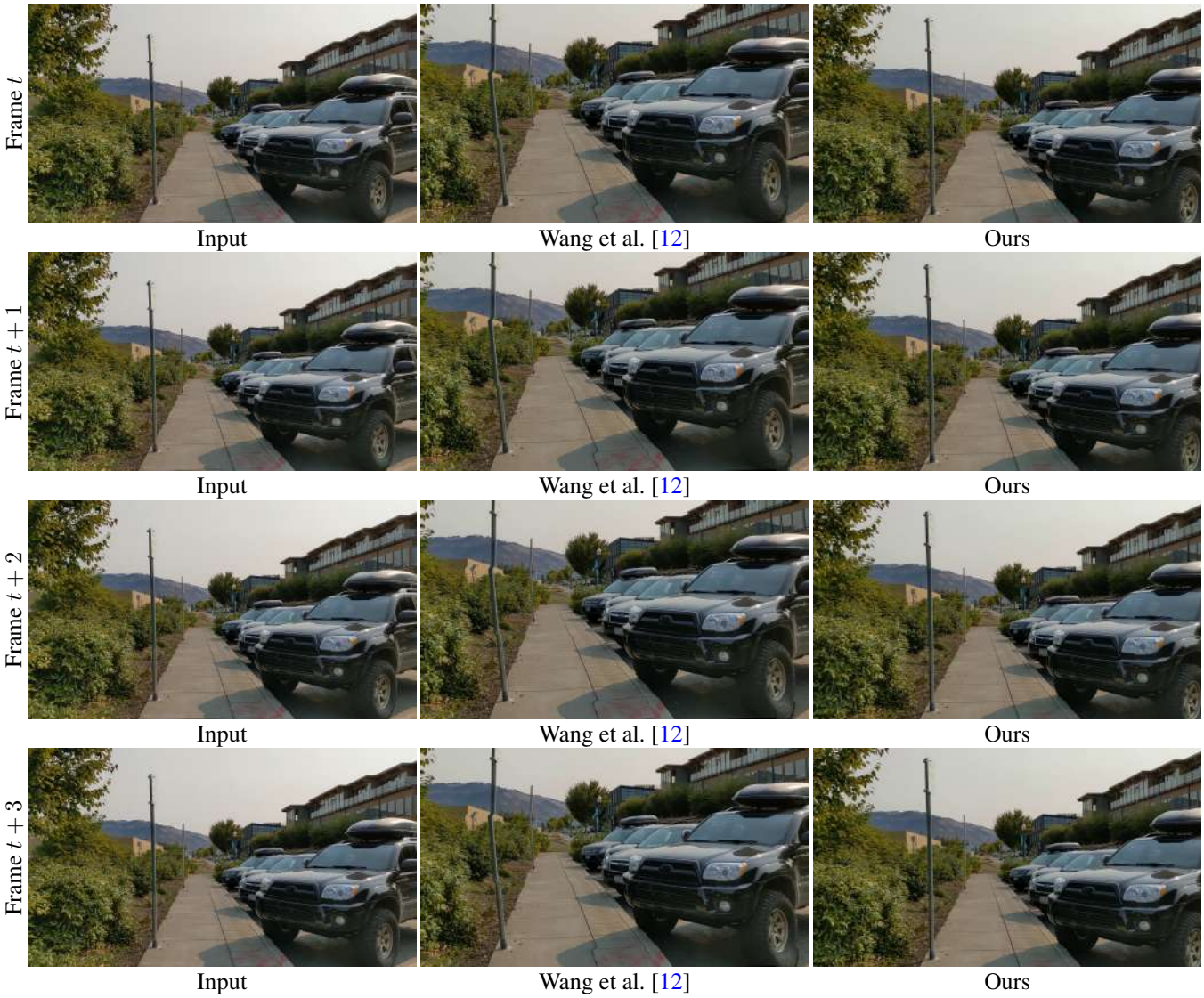

Figure 6: **Visual comparisons with Wang et al. [12].** The street light poles in Wang et al. [12]'s results are distorted, while our method maintains the scene geometry well.

Figure 7: **Visual comparisons with PWStableNet [16].** The results of PWStableNet [16] contain temporal wobbling and residual motion (see the bumping from frame t to $t + 10$).

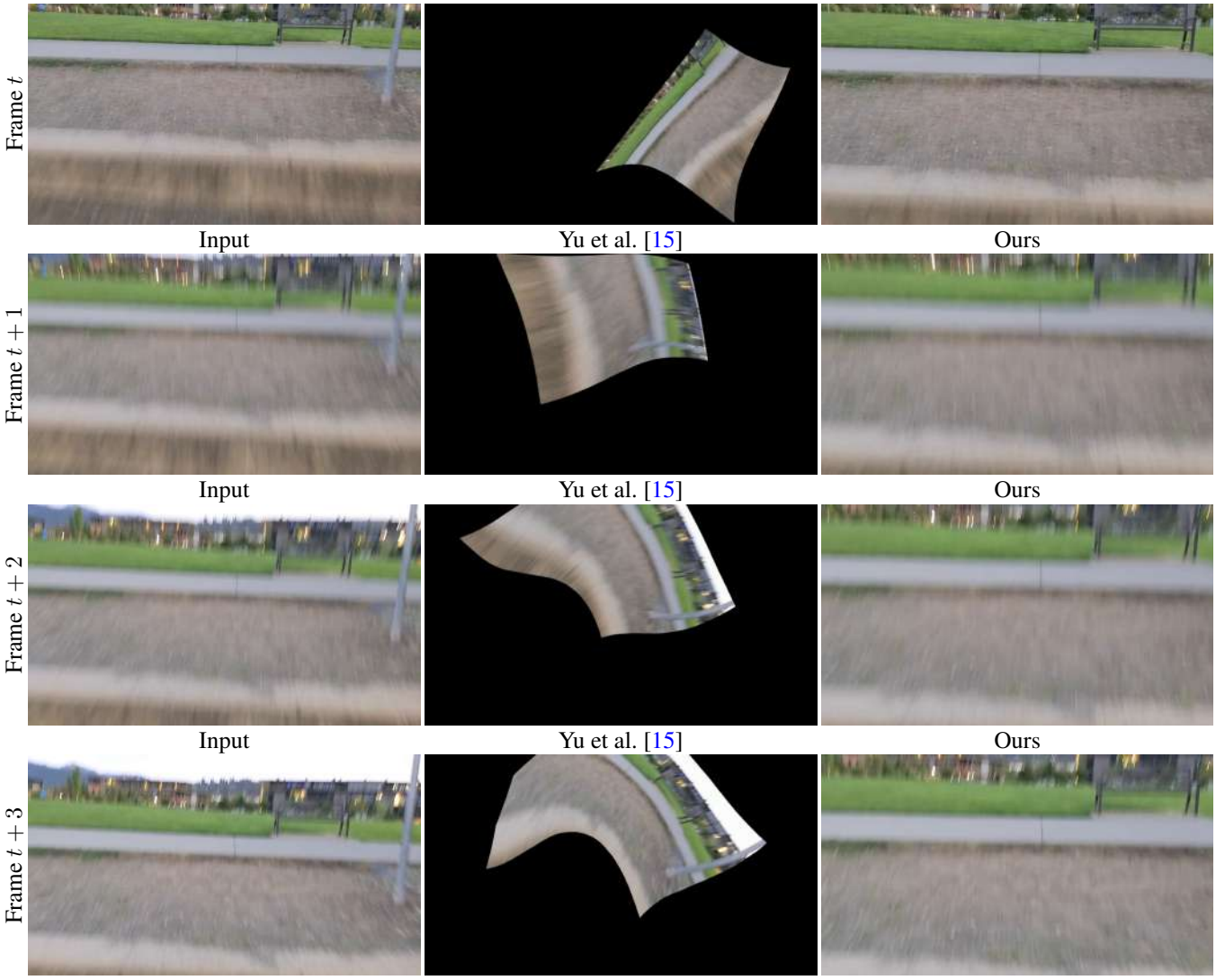

Figure 8: **Visual comparisons with Yu et al. [15].** Yu et al. [15] fails to handle a challenging RUNNING example.

Figure 9: **Visual comparisons with Choi et al. [2].** The method of Choi et al. [2] often generates ghosting and blurry results when the camera or object motion is large.