3D Multi-frame Fusion for Video Stabilization

Supplementary Material

7. Appendix

In the supplementary material, we provide the following contents:

- Additional implementation details of the feature extraction network and descriptor prediction network ϕ_{mlp};
- Quantitative evaluation on each scene;
- A 3-minute video comparing our stabilized videos with results by other methods;
- Runtime analysis of the proposed method;
- Discussion on limitations of our method.

A. Additional Implementation Details

Our feature extraction network follows IBRNet [34]. And the architecture of the descriptor prediction network ϕ_{mlp} mentioned in Sec. 3.1 is shown in Fig. 10. The inputs of ϕ_{mlp} for the sampled spatial points include: (1) input features from neighboring views, obtained by concatenating the image and feature $\mathcal{F}_t$ output by the feature extraction network ; (2) the relative viewing direction $\Delta\mathbf{d}_t$; and (3) the corrected color c'_t output by the color correction module. For predicting the density σ, we use an MLP which takes in the concatenation of the image feature $\mathcal{F}_t$ and its mean, variance to generate an intermediate feature $\mathcal{F}^*_t$ and the pooling weight ω^*_t. The density σ for each sampled spatial point is then calculated by weighted pooling $\mathcal{F}^*_t$ with ω^*_t. In order to obtain the color $\mathbf{c}$, another MLP takes the concatenation of relative viewing direction $\Delta\mathbf{d}_j$ and $\mathcal{F}^*_t$ to produce color blending weights ω^c_{t-T} for each neighboring view. Then we calculate the color $\mathbf{c}$ by weighted summing c'_t with the weights ω^c_{t-T}.

The selection of frames in the sliding window is not strictly continuous. For our experiments, after frame T, frames T+1∼T+3 are selected to ensure image quality, and frames T+10, T+15, and T+20 contribute to a sufficiently expansive field of view. It should be noted that the window size is adjustable as needed. We choose thirteen frames to balance efficiency and effectiveness, maintaining full frame in all scenarios across the three common datasets. Further details will be included in our revised version.

B. Quantitative Evaluation

Fig. 11 shows the per-scene evaluation results on the NUS [20], the Selfie [38], and the DeepStab [33] datasets. Our method and 2D-based full-frame methods [5, 24, 42] achieve the highest cropping ratio of 1 as all these methods generate no-cropping videos. On the distortion and stability metrics, our method performs on par with the state-of-the-art 3D-based method [11] and outperforms all 2D-based methods [5, 7, 20, 24, 40, 42]. In summary, the proposed method demonstrates effectiveness and robustness across diverse scenarios.

Method	Runtime
Grundmann *et al*. [7]	589ms
Bundle [20]	7264ms
Yu and Ramamoorthi [40]	6501ms
Zhao *et al*. [42]	168ms
Deep3D [11]	708ms
DIFRINT [5]	1708ms
FuSta [24]	9864ms
Ours	2836ms

Table 3. **Per-frame runtime comparison.**

C. Video Results

We provide a 3-minute video comparing our stabilized videos with results by other methods as an attachment in this supplementary material. One can download and play for a better view of our method.

D. Runtime Analysis

We test the runtime of CPU-based methods [7, 20] on a laptop with i7-10700 CPU. All GPU-based methods [5, 11, 24, 40, 42] including ours are evaluated with a single Nvidia RTX3090 GPU on a server. We perform the per-frame runtime comparison on videos with resolution of 854×480. The quantitative results are summarized in Table 3 for reference. It should be noted that this paper focuses on promoting the performance; efficiency is not our main emphasis.

All stabilizers, including the slowest baseline in Table 3, can stabilize a 100-frame video within 17 minutes without scene-specific training. However, NeRF-based methods demand per-scene training, and the runtime is impractical for video stabilization applications. For instance, LocalRF [26] and DynIBaR [14] require approximately 3 hours and 10 days, respectively, to stabilize the 100-frame video.

E. Limitations

While our method performs well in video stabilization, it has two limitations. First, our method relies on accurate prior inputs provided by the preprocessing approach [11], such as camera poses. Moreover, our method processes videos slower than some lightweight models. In the future, we plan to boost the efficiency of our method, either by devising a simpler descriptor prediction network, or by exploring engineering ways for lightweight designs.

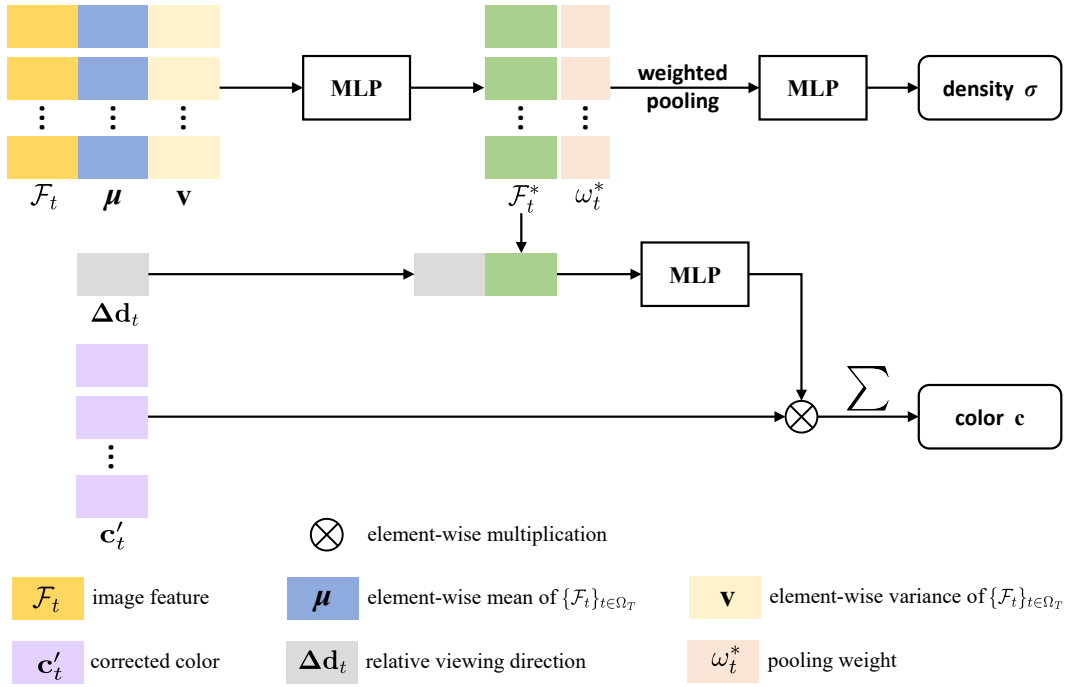

Figure 10. **Architecture of our descriptor prediction network ϕ_{mlp}.**

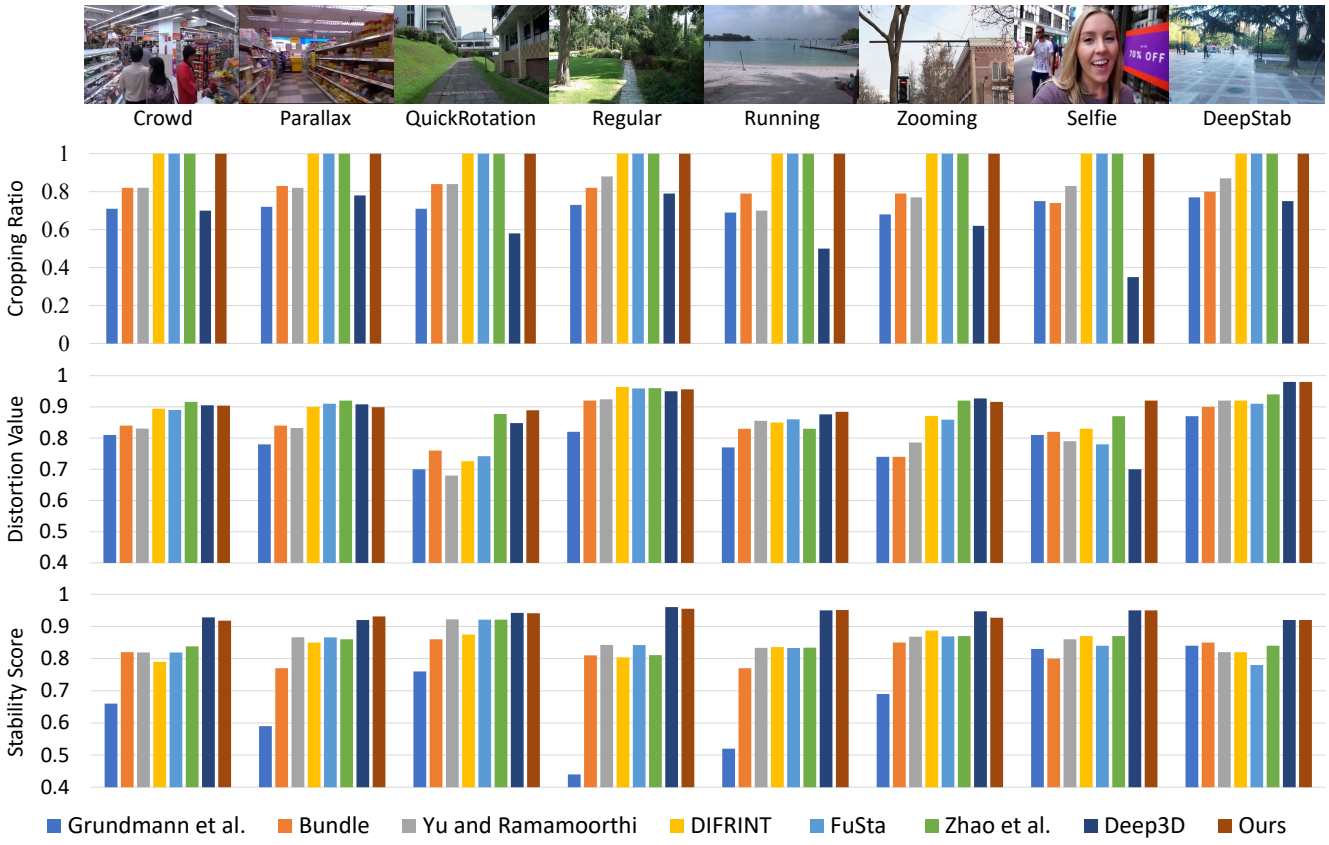

Figure 11. **Quantitative evaluation on each scene.**