This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

3D Multi-frame Fusion for Video Stabilization

Zhan Peng Xinyi Ye Weiyue Zhao Tianqi Liu Huiqiang Sun Baopu Li Zhiguo Cao*
School of AIA, Huazhong University of Science and Technology
{peng_zhan,xinyiye,zhaoweiyue,tq_liu,shq1031,zgcao}@hust.edu.cn
bpli.cuhk@gmail.com

Abstract

*In this paper, we present RStab, a novel framework for video stabilization that integrates 3D multi-frame fusion through volume rendering. Departing from conventional methods, we introduce a 3D multi-frame perspective to generate stabilized images, addressing the challenge of full-frame generation while preserving structure. The core of our RStab framework lies in **S**tabilized **R**endering (SR), a volume rendering module, fusing multi-frame information in 3D space. Specifically, SR involves warping features and colors from multiple frames by projection, fusing them into descriptors to render the stabilized image. However, the precision of warped information depends on the projection accuracy, a factor significantly influenced by dynamic regions. In response, we introduce the **A**daptive **R**ay **R**ange (ARR) module to integrate depth priors, adaptively defining the sampling range for the projection process. Additionally, we propose **C**olor **C**orrection (CC) assisting geometric constraints with optical flow for accurate color aggregation. Thanks to the three modules, our RStab demonstrates superior performance compared with previous stabilizers in the field of view (FOV), image quality, and video stability across various datasets.*

1. Introduction

With the widespread adoption of smartphones, videos have become an important medium for documenting and sharing lives. The videos captured with handheld devices often suffer from annoying shakes. To mitigate this prevalent issue, numerous researchers devote efforts to developing video stabilization algorithms. These methods typically involve three steps: camera trajectory estimation, trajectory smoothing, and stabilized frame generation.

To obtain a smooth image sequence, known as stabilized frames, early methods employ 2D-plane transformations (homography [20, 23], feature trajectories [9, 10, 21], motion vectors [18]) on single frames. However, these methods suffer from two major problems. First, these single-frame approaches may produce notable missing regions at the boundary of generated stabilized images, requiring aggressive cropping to ensure a rectangular frame for video (cropping in Fig. 1(a)), further resulting in a substantial reduction in the field of view (FOV). Second, 2D transformations could give rise to structure distortion due to the lack of 3D physical information (shear in Fig. 1(a)).

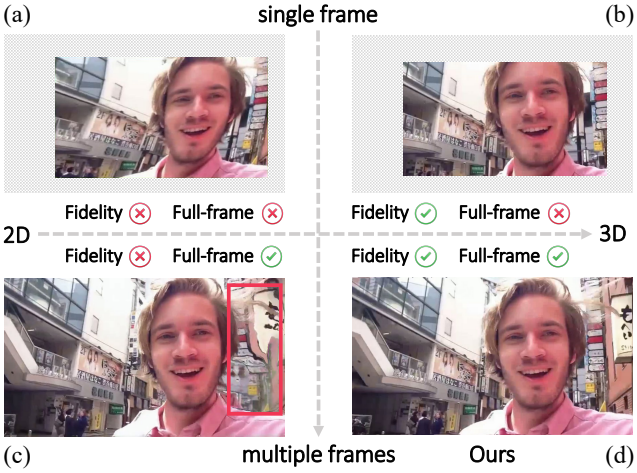

Figure 1. **Existing dilemmas and our method.** (a) and (b) exhibit cropping issues, characteristic of single-frame methods. (a) and (c) encounter difficulties in preserving structure, inherent in 2D-based approaches. Fortunately, our proposed method (d) not only mitigates distortion and artifacts but also maintains no-cropping stabilized frames.

In pursuit of the stabilized full-frame, recent 2D methods [5, 24, 42] leverage nearby frames to fill in the unseen content within the target frame. However, due to the inherent absence of physical constraints in 2D transformations, 2D-based multiple-frame methods fail to preserve the structure, especially the parallax regions (Fig. 1(c)). To obtain the structure-preserved stabilized frame, some methods [11, 17, 19, 31] leverage 3D transformations to simulate

*Corresponding author.

real-world settings, employing camera poses and epipolar constraints to ensure the image structure. However, due to limited information from a single frame, they cannot generate a full frame, as shown in Fig. 1(b). In brief, the ongoing challenge of concurrently addressing full-frame generation while preserving structure for video stabilization remains a major concern for most current research works.

To overcome the above problems, intuitively, employing multi-frame fusion with 3D transformations could offer a promising solution. However, two issues may still hinder 3D transformations from incorporating information from neighboring frames. First, since view changes induce geometric deformation, the incorporated information from nearby frames may be inconsistent, suggesting that image blending, e.g., averaging, may lead to distortion. Second, videos feature dynamic objects across frames, which cannot be adequately modeled by 3D constraints. The direct aggregation of information from nearby frames with 3D projection results in a noticeable blur (refer to the experiments).

Motivated by the above insights and analysis, we propose a video stabilization framework termed RStab for integrating multi-frame fusion and 3D constraints to achieve full-frame generation and structure preservation. Specifically, we propose **S**tabilized **R**endering (SR), a 3D multi-frame fusion module using volume rendering. Instead of simple image blending, SR employs both color and feature space to fuse nearby information into spatial descriptors for the scene geometry, such as volume densities of spatial points. Visible points usually come with high volume densities, exhibiting consistent textures in their projections across frames. The observation suggests that points with higher consistency in aggregating information exhibit higher volume densities, implying a greater contribution to the final rendered color.

To mitigate the impacts of dynamic regions, we propose **A**daptive **R**ay **R**ange (ARR) and **C**olor **C**orrection(CC) modules. The introduction of multi-frame depth priors in ARR constrains the sampling range for spatial points around the surface of objects. A narrow sampling range around the surface decreases the risk of projecting spatial points onto dynamic regions, thereby suppressing the inconsistent information aggregation induced by the dynamic objects. Despite ARR, colors are sensitive to projection inaccuracy, indicating a narrow range is insufficient. Hence, we design CC to refine the projection for color aggregation. The core of CC lies in assisting geometry constraints with optical flow, which matches pixels with similar textures containing the color information.

By applying the three modules, RStab demonstrates the ability of full-frame generation with structure preservation (Fig. 1(d)) and outperforms all previous video stabilization algorithms in FOV, image quality, and video stability across various datasets. In summary, our key contributions are as follows:

- We present a novel 3D multi-frame fusion framework for video stabilization to render full-frame stabilized images with structure preservation.
- We propose Stabilized Rendering, which fuses multiple frames in both color and feature space. We augment Stabilized Rendering with the introduction of the Adaptive Ray Range module and Color Correction module, enhancing its capacity to address dynamic regions.
- Our video stabilization framework, RStab, demonstrates state-of-the-art (SOTA) performance across various datasets.

2. Related Work

2D-based Video Stabilization. 2D video stabilization algorithms model camera trajectory and generate stabilized frames through transformations on a 2D plane, including homography [6, 7, 20, 23, 40], feature trajectories [9, 10, 21, 38], motion vectors [16, 18, 22, 36], and dense flow fields [4, 5, 24, 40, 41]. Early methods [6, 7] estimate global transformations, which proved inadequate for handling complex camera effects such as the parallax effect. Certain approaches estimate multiple local motions [20, 22] or pixel-wise warping field [36, 40, 41] for a single image, offering some relief for the challenges encountered by global transformation methods. However, due to the limited information from a single frame, these methods may result in missing content in the stabilized video. To address this, some methods [5, 24, 42] fuse information from multiple neighboring frames, enabling full-frame generation. Despite achieving a full frame, the 2D transformations lack real-world physical constraints, leading to challenges in preserving image structure.

3D-based Video Stabilization. 3D-based video stabilizers model 3D camera trajectory and stabilize frames with epipolar projection. Some methods [11, 17] rely on the video itself, warping images instructed by projection while preserving content. Others integrate specialized hardware, such as depth cameras [19], light field cameras [31], gyroscopes [30], and IMU sensors [12], to assist with scene geometry. Both kinds of stabilizers estimate the physical motion of the real world and introduce 3D constraints in warping, benefiting stability and structure preservation. However, relying on a single frame, 3D-based video stabilizers have a limited field of view. To mitigate the issue, in this paper, we extend single-frame to multi-frame in 3D space for video stabilization.

Neural Rendering. As a significant work in view synthesis, NeRF[15] attains photorealistic synthesized images through implicit volumetric representation and volume rendering. It combines multi-view information, leveraging 3D geometric constraints and pixel-wise rendering to generate high-quality images without missing content from novel

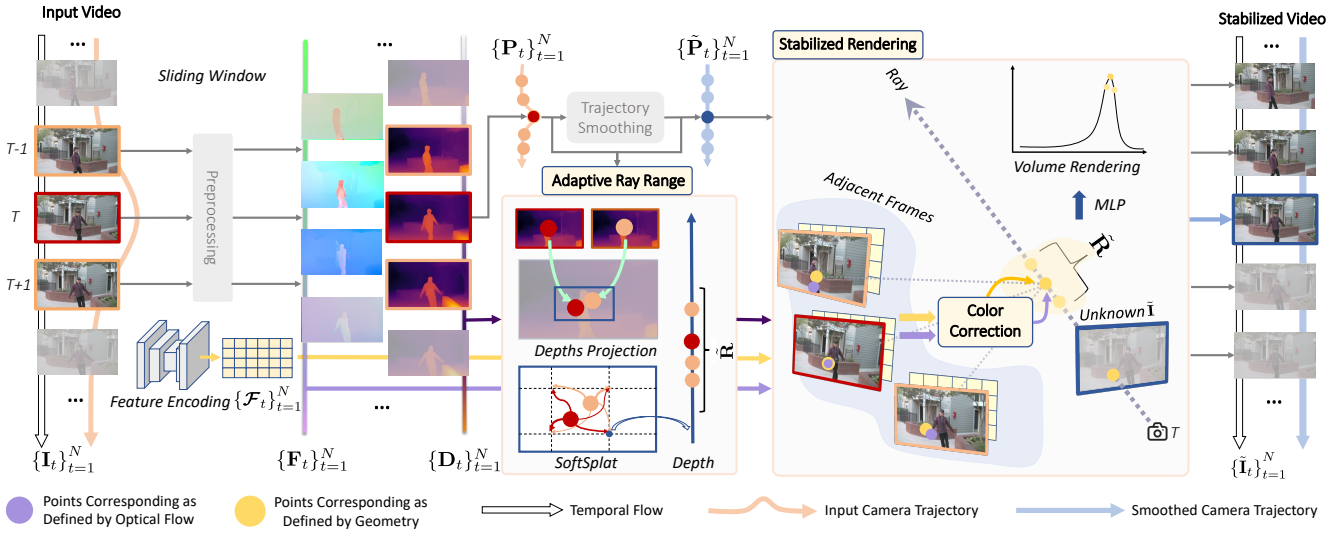

Figure 2. **Overview of our framework.** (1) Given input frames $\{\mathbf{I}_t\}_{t=1}^N$ with a shaky trajectory $\{\mathbf{P}_t\}_{t=1}^N$, our purpose lies in rendering stabilized video sequence $\{\tilde{\mathbf{I}}_t\}_{t=1}^N$ with smoothed trajectory $\{\tilde{\mathbf{P}}_t\}_{t=1}^N$. Here, the input trajectories $\{\mathbf{P}_t\}_{t=1}^N$ derive from preprocessing, while the smoothed trajectories $\{\tilde{\mathbf{P}}_t\}_{t=1}^N$ are generated using a Trajectory Smoothing module. (2) In addition to $\{\mathbf{P}_t\}_{t=1}^N$, depth maps $\{\mathbf{D}_t\}_{t=1}^N$ and optical flow $\{\mathbf{F}_t\}_{t=1}^N$ can be obtained during preprocessing. We aggregate $\{\mathbf{D}_t\}_{t=1}^N$ into the ray range $\{\tilde{\mathbf{R}}_t\}_{t=1}^N$ using the Adaptive Ray Range module. The ray range $\{\tilde{\mathbf{R}}_t\}_{t=1}^N$, along with $\{\mathbf{F}_t\}_{t=1}^N$ and the smoothed trajectory $\{\tilde{\mathbf{P}}_t\}_{t=1}^N$, serves as inputs to the Stabilized Rendering module. Conducting Stabilized Rendering, enhanced by the Color Correction module, we fuse the input frames $\{\mathbf{I}_t\}_{t=1}^N$ and their features $\{\mathcal{F}_t\}_{t=1}^N$ to render the stabilized video sequence $\{\tilde{\mathbf{I}}_t\}_{t=1}^N$.

viewpoints. While NeRF-based methods [1, 2, 13, 25, 28] produce impressive synthesized image quality, its limitation in per-scene training hampers its direct application in video stabilization. Certain approaches [3, 15, 32, 34, 35, 37] strive to improve the generalization of NeRF, but they are not inherently well-suited for video stabilization tasks. Some recent methods [14, 26] attempt to apply techniques in NeRF to stabilize videos, these approaches inherit the limitations of the vanilla NeRF, necessitating retraining for each specific scene. Inspired by generalized rendering technologies from IBRNet[34] and ENeRF[15], which utilize multi-view images and associated features to predict radiance fields, we further propose the Stabilized Rendering. Stabilized Rendering, enhanced by the proposed Adaptive Ray Range module and Color Correction module, extends the volume rendering technique to video stabilization.

3. Method

Our pipeline is shown in Fig. 2. Given a shaky frame sequence $\{\mathbf{I}_t\}_{t=1}^N$ of length N, our objective is to generate a stabilized sequence $\{\tilde{\mathbf{I}}_t\}_{t=1}^N$. For preprocessing of $\{\mathbf{I}_t\}_{t=1}^N$, we estimate optical flow $\{\mathbf{F}_t\}_{t=1}^N$, depth maps $\{\mathbf{D}_t\}_{t=1}^N$, and camera trajectory $\{\mathbf{P}_t\}_{t=1}^N$. With $\{\mathbf{D}_t\}_{t=1}^N$, $\{\mathbf{P}_t\}_{t=1}^N$ and smoothed camera trajectory $\{\tilde{\mathbf{P}}_t\}_{t=1}^N$ as input, the Adaptive Ray Range module aggregates multi-view depth maps into the ray ranges $\{\tilde{\mathbf{R}}_t\}_{t=1}^N$. Guided by the ranges, Stabilized Rendering enhanced by the Color Correction module generates stabilized video sequence $\{\tilde{\mathbf{I}}_t\}_{t=1}^N$ through fusing the input frames $\{\mathbf{I}_t\}_{t=1}^N$ and feature maps $\{\mathcal{F}_t\}_{t=1}^N$ obtained through feature extraction network.

We start with preprocessing a sequence of input frames $\{\mathbf{I}_t\}_{t=1}^N$ to estimate associated depth maps $\{\mathbf{D}_t\}_{t=1}^N$ and camera trajectory $\{\mathbf{P}_t\}_{t=1}^N$. These depth maps and camera poses are employed for camera trajectory smoothing. In our pursuit of consistent and smooth camera trajectories, we harness the flexibility of the Gaussian smoothing function: $\{\tilde{\mathbf{P}}_t\}_{t=1}^N = \phi_{sm}(\{\mathbf{P}_t\}_{t=1}^N)$, where ϕ_{sm} represents the Gaussian smoothing filter, offering adjustable parameters for both the smoothing window and stability. These parameters can be fine-tuned to meet specific requirements and constraints. In Sec. 3.1, we elaborate on rendering a stabilized image with its neighboring frames through Stabilized Rendering. Due to dynamic regions, the conventional 3D-constraint-based rendering fails to adequately represent the geometry. Differing from the conventional rendering, Sec. 3.2 introduces the utilization of depth priors to constrain the sampling range of spatial points around potential geometries, such as the area around the surface of objects. Additionally, in Sec. 3.3, we discuss refining projecting inaccuracy to ensure consistent local color intensities.

Stabilizing a video involves rendering a image sequence $\{\tilde{\mathbf{I}}_t\}_{t=1}^N$ with corresponding stabilized poses $\{\tilde{\mathbf{P}}_t\}_{t=1}^N$. In

practice, we adopt a sliding window strategy for frame-by-frame rendering stabilized video. For clarity, we illustrate the rendering process with a single target camera pose $\tilde{\mathbf{P}}$ at the timestamp T and its temporal neighborhood Ω_T.

3.1. Stabilized Rendering

Stabilized Rendering is a multi-frame fusion module founded on epipolar constraints which fuses input images and feature maps to render a stable, uncropped video sequence. Considering a pixel $\tilde{\mathbf{x}}$ situated in the stabilized image $\tilde{\mathbf{I}}$ under a specific target camera pose $\tilde{\mathbf{P}}$, we sample L spatial points sharing projection situation $\tilde{\mathbf{x}}$. These sampled points span depth $\{\tilde{d}_i\}_{i=1}^{L}$ distributed along the ray with sampling range, denoted as $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$. We project $\tilde{\mathbf{x}}$ at depth $\tilde{d}_i$ onto the neighboring input frames $\{\mathbf{I}_t\}_{t\in\Omega_T}$ at corresponding positions $\{\mathbf{x}_t^i\}_{t\in\Omega_T}$ by

$$\mathbf{x}_t^i = \mathbf{K}\mathbf{P}_t\tilde{\mathbf{P}}^{-1}\tilde{d}_i\mathbf{K}^{-1}\tilde{\mathbf{x}}\,, \tag{1}$$

where $\mathbf{K}$ represents the camera intrinsic parameters shared by all frames in a video and $i \in (0, L]$. With the projected points $\{\mathbf{x}_t^i\}_{t\in\Omega_T}$, we aggregate features $\{\boldsymbol{\mathcal{F}}_t(\mathbf{x}_t^i)\}_{t\in\Omega_T}$ in neighboring frames to predict the volume density σ_i for the spatial point by

$$\sigma_i = \phi_{mlp}(\{\boldsymbol{\mathcal{F}}_t(\mathbf{x}_t^i)\}_{t\in\Omega_T})\,, \tag{2}$$

where ϕ_{mlp} is a Multiple Layer Perceptron (refer to Supp. for details). Eq. 2 is contingent upon the consistency among features. Specifically, if a sampled spatial point aligns with the ground geometry, the multi-view features of projected points would be similar. This condition establishes scene-independent geometric constraints. When considering the associated color $\mathbf{c}_i$, a conventional method is a linear combination for aggregation:

$$\mathbf{c}_i = \sum_{t\in\Omega_T} \omega_{t-T}\mathbf{I}_t(\mathbf{x}_t^i)\,,\ \sum_{t\in\Omega_T}\omega_{t-T} = 1\,, \tag{3}$$

where ω_{t-T} represents adaptable parameters determined by the geometric characteristics, such as the volume density σ_i. Since the establishment of $\mathbf{c}_i$ solely relies on input frames, it is training-free to accommodate unforeseen scenes. In volume rendering, the set $\{\mathbf{c}_i, \sigma_i\}_{i=1}^{L}$, describing spatial points along the same ray, determine the color intensity of $\tilde{\mathbf{x}}$ by

$$\begin{aligned} \tilde{\mathbf{I}}(\tilde{\mathbf{x}}) &= \sum_{i=1}^{L} A_i(1 - exp(-\sigma_i))\mathbf{c}_i, \\ A_i &= exp\left(-\sum_{j=1}^{i-1}\sigma_i\right). \end{aligned} \tag{4}$$

In Stabilized Rendering, Eqs. 1 imposes epipolar constraints on features and colors warped from multiple neighboring frames. Eqs. 2 & Eqs. 3 aggregate the multi-frame information into spatial descriptors $\{\mathbf{c}_i, \sigma_i\}_{i=1}^{L}$, and

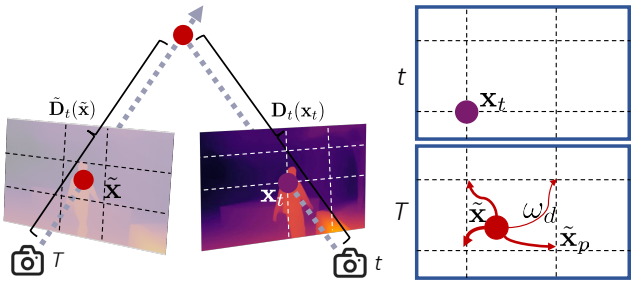

Figure 3. **Illustration of depth projection and splatting.** Left: The depth projection involve lifting a pixel $\mathbf{x}_t$ to 3D space using the estimated depth $\mathbf{D}_t(\mathbf{x}_t)$ and projecting to the sub-pixel $\tilde{\mathbf{x}}$. The depth of $\tilde{\mathbf{x}}$ can be calculated and denoted as $\tilde{\mathbf{D}}_t(\tilde{\mathbf{x}})$. Right: As $\tilde{\mathbf{x}}$ is not precisely projected onto a pixel coordinate, we convert its depth to adjacent pixels, e.g. $\tilde{\mathbf{x}}_p$, with a distance-associated weight ω_t.

Eqs. 4 renders stabilized images utilizing these descriptors for each pixel. Epipolar constraints guarantee the structure preservation and per-pixel rendering guarantees full-frame generation. However, the effectiveness of the aforementioned process highly depends on the ray range $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$ guiding the sampling. If $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$ is not distributed near the surface of objects, the model may aggregate incorrect features into inferior descriptors and diminish rendering quality. The forthcoming section will introduce how to adaptively define the ray range $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$ to avoid the issue above.

3.2. Adaptive Ray Range

Eq. 4 of Stabilized Rendering highlights the dependence of the final color intensity of $\tilde{\mathbf{I}}(\tilde{\mathbf{x}})$ on the color $\mathbf{c}_i$ of the 3D point where the ray hits the object for the first time. It indicates that ray ranges around the ground geometry for the sampling process will benefit scene representation. A direct method to define the ray range entails treating the sequence of frames as a static scene: estimating the coarse geometry of each ray and rendering through spatial points sampled from re-defined fine ranges, such as [15, 34]. We argue that the effectiveness of the coarse-to-fine ray range relies on the geometry estimation grounded in epipolar constraints. However, dynamic regions, violating epipolar constraints, make the defined range unreliable.

To tackle this challenge, we turn to the task of depth estimation. The depth model [11] employs optical flow to impose constraints on dynamic scenes. As optical flow relies on feature matching rather than epipolar constraints, it matches points with features rather than epipolar constraints, showcasing insensitivity to dynamic regions. Consequently, the estimated depth maps derived from this depth model are less susceptible to interference from dynamic objects. We propose to define an adaptive range with pre-estimated neighboring depth maps $\{\mathbf{D}_t\}_{t\in\Omega_T}$. In particular, we construct the range utilizing the mean and variance

w/o Temporal Weights Ours

Figure 4. **The effect of temporal weights.** The introduction of temporal weights can mitigate distortion.

of aggregated depth maps from nearby frames.

As illustrated in the left part of Fig. 3, we project $\mathbf{x}_t$ in the neighboring frame with pose $\mathbf{P}_t$ at the depth $\mathbf{D}_t(\mathbf{x}_t)$ onto sub-pixel $\tilde{\mathbf{x}}$ of the stabilized frame with pose $\tilde{\mathbf{P}}$ according to the inverse of Eq. 1. However, as sub-pixel $\tilde{\mathbf{x}}$ is not precisely projected onto a specific pixel coordinate, direct utilization of $\tilde{\mathbf{D}}_t(\tilde{\mathbf{x}})$ to estimate ray ranges for pixels is not feasible. To overcome this limitation, a splatting method [29] is employed, as illustrated in the right part of Fig. 3, converting $\tilde{\mathbf{D}}_t(\tilde{\mathbf{x}})$ in the following manner:

$$\tilde{\mathbf{D}}_t(\tilde{\mathbf{x}}_p) = \frac{\sum_i w_d \tilde{\mathbf{D}}_t(\tilde{\mathbf{x}}_i)}{\sum_i w_d}, w_d = \prod(\mathbf{1} - |\tilde{\mathbf{x}}_p - \tilde{\mathbf{x}}_i|), \quad (5)$$

where $\tilde{\mathbf{x}}_p$ is a pixel and $\tilde{\mathbf{x}}_i$ is the i-th sub-pixel $\tilde{\mathbf{x}}$ around $\tilde{\mathbf{x}}_p$ satisfying the condition $|\tilde{\mathbf{x}}_p - \tilde{\mathbf{x}}_i| \in (0,1)^2$, $\prod(\cdot)$ suggests an element-wise multiplication in a vector, and ω_d is distance-associated weights.

Given $\{\mathbf{D}_t\}_{t\in\Omega_T}$, we obtain corresponding $\{\tilde{\mathbf{D}}_t\}_{t\in\Omega_T}$ on the stabilized frame through the project-splat process above. An intuitive approach involves directly calculating the mean $\mathbf{M}$, variance $\mathbf{S}$, and determining the sampling ray range as $\mathbf{R} = [\mathbf{M} - \mathbf{S}, \mathbf{M} + \mathbf{S}]$. However, in the aforementioned depth project-splat process, depth maps further from the timestamp T are less reliable. Treating all depth maps equally can result in an inaccurate sampling ray range $\mathbf{R}$, leading to a decrease in the image quality (the left part of Fig. 4). This observation prompts the introduction of a weighted mean and variance as:

$$\tilde{\mathbf{M}} = \sum_{t\in\Omega_T} \omega_t \tilde{\mathbf{D}}_t, \tilde{S} = \sqrt{\sum_{t\in\Omega_T} \omega_t (\tilde{\mathbf{D}} - \tilde{\mathbf{M}})^2}, \quad (6)$$

where ω_t is the temporal weighting coefficient, assigning a higher weight to the frame closer to the stabilized frame temporally and vice versa, as defined by

$$\omega_t = \frac{e^{\lambda(t-T)}}{\sum_{t\in\Omega_T} e^{\lambda(t-T)}}, \quad (7)$$

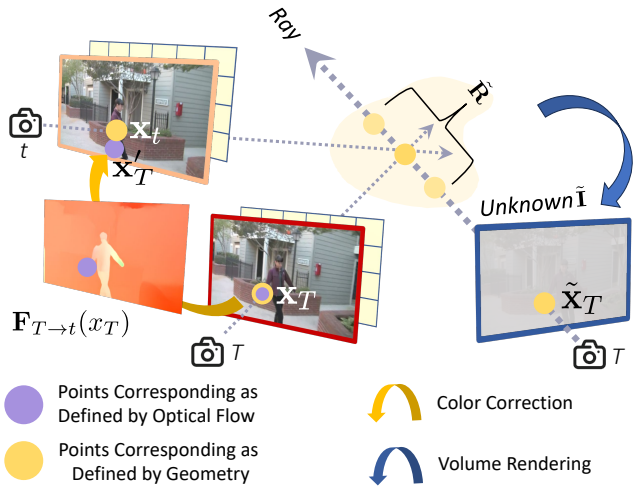

Figure 5. **Illustration of Color Correction module.** Firstly, we project a pixel $\tilde{\mathbf{x}}_T$ from the target stabilized frame onto corresponding $\mathbf{x}_T$ of the input frame at the same timestamp T. Secondly, we obtain feature matching of $\mathbf{x}_T$ in the input frame at timestamps t using optical flow $\mathbf{F}_{T\to t}(\mathbf{x}_T)$. As geometric constraints alone are insufficient for modeling dynamic regions, we aggregate precise color by correcting the geometric projected position $\mathbf{x}_t$ to the optical-flow refined position $\mathbf{x}'_t$.

where λ is a hyperparameter. Subsequently, ray ranges for the stabilized frame are denoted as $\tilde{\mathbf{R}} = \left[\tilde{\mathbf{M}} - \tilde{\mathbf{S}}, \tilde{\mathbf{M}} + \tilde{\mathbf{S}}\right]$. and can be employed for sampling L points along each ray during the rendering process. As illustrated in the right part of Fig. 4, the Adaptive Ray Range module with temporal weighted ranges yields more favorable rendering results.

The Adaptive Ray Range module provides a ray range $\tilde{\mathbf{R}}$ around the ground geometry guiding points sampling and benefiting volume density σ_i prediction. Although the guidance of $\tilde{\mathbf{R}}$ mitigates the interference of dynamic objects, the challenge of dynamic objects goes beyond this. According to Eq. 4, the color intensity $\mathbf{c}_i$ is another factor influencing rendering quality and affected by dynamic regions as well.

3.3. Color Correction

Color intensity, denoted as $\mathbf{c}_i$, exhibits a strong dependence on geometric constraints, akin to volume density σ_i. However, density is predicted from the feature maps with their receptive fields, thereby exhibiting a certain tolerance to projection inaccuracy. In contrast, color intensity is derived from the linear combination of colors warped from multiple views, accentuating the sensitivity of colors to projection inaccuracy. Despite the Adaptive Ray Range module offers a correction for projection with geometric constraints, it is inadequate for accurate color aggregation (refer to the experiments). Rather than solely concentrating on refining geometric constraints, we propose to assist these constraints with optical flow. Optical flow, relying on feature similar-

Method		NUS dataset			Selfie dataset			DeepStab dataset		
		C↑	D↑	S↑	C↑	D↑	S↑	C↑	D↑	S↑
Grundmann *et al*. [7]	2D	0.71	0.76	0.82	0.75	0.81	0.83	0.77	0.87	0.84
Bundle [20]	2D	0.81	0.78	0.82	0.74	0.82	0.80	0.80	0.90	0.85
Yu and Ramamoorthi [40]	2D	0.85	0.81	0.86	0.83	0.79	0.86	0.87	0.92	0.82
DIFRINT [5]	2D	**1.00**	0.87	0.84	**1.00**	0.78	0.84	**1.00**	0.91	0.78
FuSta [24]	2D	**1.00**	0.87	0.86	**1.00**	0.83	0.87	**1.00**	0.92	0.82
Zhao *et al*. [42]	2D	**1.00**	0.90	0.87	**1.00**	0.87	0.87	**1.00**	0.94	0.84
Deep3D [11]	3D	0.66	0.90	**0.94**	0.35	0.70	**0.95**	0.75	**0.98**	**0.92**
Ours	3D	**1.00**	**0.91**	**0.94**	**1.00**	**0.92**	**0.95**	**1.00**	**0.98**	**0.92**

Table 1. **Quantitative results on the NUS [20], the Selfie [38], and the DeepStab [33] datasets.** We evaluate our method against baselines using three standard metrics: Cropping Ratio(C), Distortion Value(D), Stability Score(S). The best results are **bolded** and second-best results are hightlighted by underline.

ities, matches pixels with similar textures containing color information. It implies that utilizing optical flow to refine the projection can enhance color accuracy.

Specifically, we focus on the input frame at T, which adheres to epipolar constraints with the target stabilized frame at T. As shown in Fig. 5, we employ $\mathbf{I}_T$ as a reference to correct the projection points on the neighboring frame $\mathbf{I}_t$ with optical flow. According to Eq. 1, we project a point $\tilde{\mathbf{x}}_T$ from the stabilized pose $\tilde{\mathbf{P}}_T$ onto the $\mathbf{x}_T$ of $\mathbf{P}_T$, the flow-associated points $\mathbf{x}'_t$ can be expressed as

$$\mathbf{x}'_t = \mathbf{x}_T + \mathbf{F}_{T\to t}(\mathbf{x}_T), \tag{8}$$

where $\mathbf{F}_{T\to t}$ represents the optical flow from $\mathbf{I}_T$ to $\mathbf{I}_t$. By applying the same procedure to frames in the temporal neighborhood Ω_T, we substitute the $\mathbf{x}_t$ in Eq. 3 with $\mathbf{x}'_t$.

3.4. Implementation Details

In our implementations, a pre-trained model from Deep3D [11] is employed to generate depth prior for the Adaptive Ray Range module and optical flow for Color Correction. Frames neighboring the timestamp T are symmetrically distributed, and the length of the set Ω_T is fixed to 13. For the Adaptive Ray Range module, the temporal weighting coefficient ω_i is calculated with $\lambda = 0.5$, and we choose $L = 3$ for uniform spatial points sampling along each ray.

Loss function. During training, we sample rays on all images randomly and minimize the mean squared error between the rendered color and corresponding ground truth:

$$\mathcal{L} = \sum_{\mathbf{x}\in\mathcal{X}} \left|\left|\tilde{\mathbf{I}}(\mathbf{x}) - \mathbf{I}_{gt}(\mathbf{x})\right|\right|_2^2, \tag{9}$$

where $\mathbf{I}_{gt}$ is the corresponding ground truth and $\mathcal{X}$ is the set of pixels sampled from all images in each training batch.

Training details. We follow the training setting of IBRNet [34] to train our model on LLFF [27] and IBRNetCollected [34] including high-quality natural images with accurate camera poses. Our model is trained on an RTX3090 GPU using the Adam optimizer[8]. We set the base learning rates for the feature extraction network and MLP to $1e^{-3}$ and $5e^{-4}$, respectively, which decay exponentially throughout the optimization process. Typically, the model converges after approximately 200k iterations, and the entire training process takes about a day to complete.

4. Experiments

4.1. Quantitative Evaluation

Baselines. We choose various video stabilization algorithms as the baselines, including Grundmann *et al*. [7], Liu *et al*. [20], Wang *et al*. [33], Yu and Ramamoorthi [39, 40], DIFRINT [5], FuSta [24], Zhao *et al*. [42], and Deep3D [11]. For comparisons, we use the official-provided videos or videos generated by official implementations with default parameters or pre-trained models.

Datasets. We choose three datasets with different characteristics for evaluations: (1) The NUS [20] dataset comprises 144 videos, categorized into six different scenes: Regular, Running, Crowd, Parallax, QuickRotation, and Running, (2) the Selfie dataset [38] contains 33 video clips featuring frontal faces with large camera motion, (3) and the DeepStab dataset [33] includes 61 high-definition videos.

Metrics. We assess the performance of the stabilizers using three standard metrics widely employed in previous methods [5, 20, 24, 39, 40]: (1) *Cropping Ratio*: This metric measures the remaining image area after cropping the non-content pixels. (2) *Distortion Value*: This metric quantifies the anisotropic scaling of the homography matrix between the input and output frames. (3) *Stability Score*: This metric assesses the stability of the stabilized video by assessing the ratio of low-frequency motion energy to the total energy. All three metrics range from 0 to 1, with higher values indicating better performance.

Results on the NUS dataset. Our evaluation on the NUS

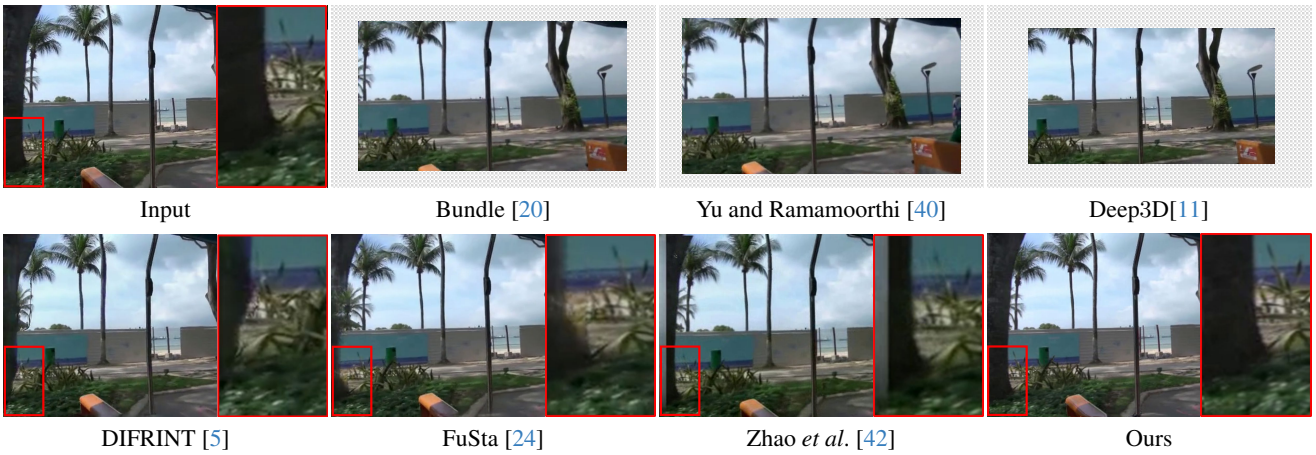

Figure 6. **Visual comparison of different methods.** Contrasting with the baselines in the first row, our method successfully accomplishes full-frame generation. In the second row, while these baselines achieve full-frame generation, they fall short in preserving structure; for instance, in the bottom-left region, the tree trunks are missing in their stabilized images. Please refer to our supplementary material for video comparisons with baselines.

dataset [20] is detailed on the left side of Table 1, where our stabilization method excels notably in both stability and distortion reduction when compared to 2D-based methods. This success is attributed to our accuracy in constructing camera trajectories and geometry. In contrast to 3D methods, our approach stands out by leveraging information from multiple input frames, achieving an average cropping ratio of 1. This indicates the effectiveness of our method in full-frame generation across the diverse scenes in the NUS dataset, which is widely acknowledged as a robust benchmark for video stabilization algorithms.

Results on the Selfie dataset. We present the results on the Selfie dataset [38] in the middle of Table 1. It's crucial to highlight that this dataset is characterized by large camera motions and extensive dynamic regions, posing challenges for video stabilization algorithms. Observing the results, a decrease is evident for most algorithms compared to their performance on the NUS dataset. Traditional 3D methods, in particular, experience a significant decline. In contrast, our method consistently delivers the best performance on the Selfie dataset. The performance shows the effectiveness of our algorithm in handling extreme scenes.

Results on the DeepStab dataset. The right side of Table 1 showcases the average scores on the DeepStab dataset [33]. Notably, the videos in this dataset are of higher resolution than NUS and Selfie, specifically 720p, aligning with the common resolutions of modern devices. Despite the high distortion values across all stabilizers due to the simplicity of this dataset, our approach consistently demonstrates superior performance. This result suggests that our method is well-suited for handling high-definition videos, further emphasizing its applicability for contemporary video stabilization challenges.

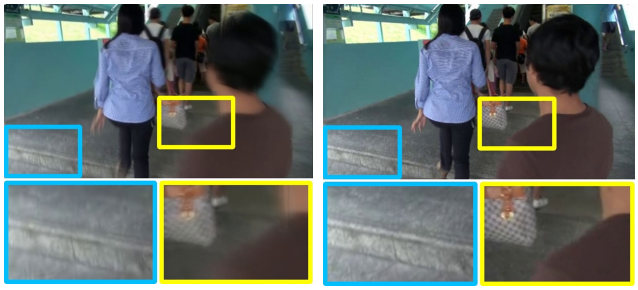

Figure 7. **Quilitative ablation of Stabilized Fusion.** Absence of Stabilized Fusion results in noticeable blurs in both static and dynamic regions.

4.2. Qualitative Analysis

Visual comparisons of our method and state-of-the-art stabilizers is shown in Fig. 6. Many methods [11, 20, 40] apply aggressive cropping, as evident from the grey checkerboard regions. Comparing the bottom-left region of each image in Fig. 6 below with the top-left input, it's clear that our method suffers from fewer visual artifacts.

5. Ablation Study

We conduct ablation studies to analyze the effectiveness of the proposed modules, including **S**tabilized **R**endering (SR), the **A**daptive **R**ay **R**ange module (ARR), and **C**olor **C**orrection module (CC). Our evaluations focus on the Crowd scene within the NUS dataset [20], chosen for its dynamic objects and diverse scenes. We choose Distortion values and PSNR as evaluation metrics. Distortion Value measures the pose-independent structure quality of

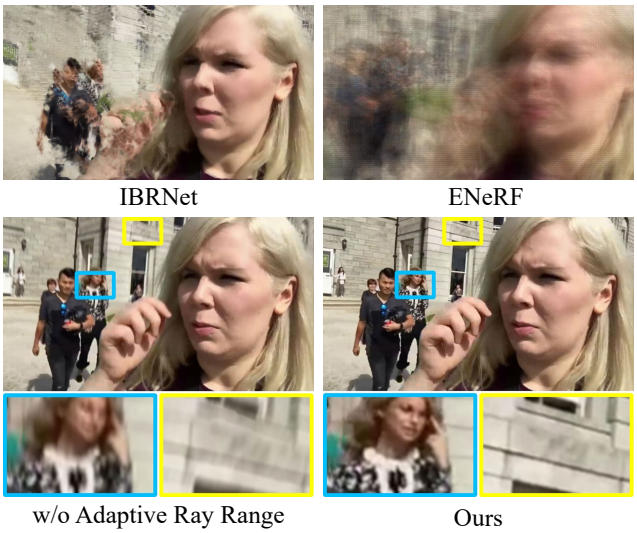

Figure 8. **Quilitative ablation of different range strategies.** Among the range strategies examined, only our Adaptive Ray Range module can address distortion in image structure.

images with stabilized poses. Additionally, PSNR is employed to evaluate the pixel-level performance of our model in rendering image details. As real images with stabilized poses are unavailable, we render images with the input pose to derive PSNR.

Why needs Stabilized Rendering. We conduct experiments to demonstrate the necessity of SR, which fuses features and colors in 3D space. One straightforward strategy replacing SR for fusing multiple frames is image blending. It warps nearby frames into the stabilized view and averages these images. However, as illustrated in the left part of Fig. 7, image blending leads to noticeable blur in both static regions (the stairs) and dynamic regions (the handbag and the shoulder). Comparing Row 4 and Row 3 in Table 1, the notable decreases in distortion value and PSNR align with the observation in Fig. 7. It demonstrates SR, our 3D multi-frame fusion module using volume rendering, can enhance the structural quality of stabilized images.

Importance of Adaptive Ray Range. We compare various range strategies to affirm the importance of ARR: (1) IBRNet [34] and ENeRF [15] employ coarse-to-fine range strategy, and (2) we adopt even sampling of 128 points following setting of IBRNet as a substitution for ARR. However, as shown in Fig. 8, none of these strategies achieve favorable results. Without the sampling range defined by ARR, the methods above are forced to aggregate points sampled over a large range, increasing the risk of projecting spatial points onto dynamic regions. Due to the violation of epipolar constraints, dynamic regions introduce incorrect features and colors to the aggregation of descriptors and lead to distortion of the structure. As shown in Row 1,2,3,5 of Table 1, ARR proves effective in preserving structure.

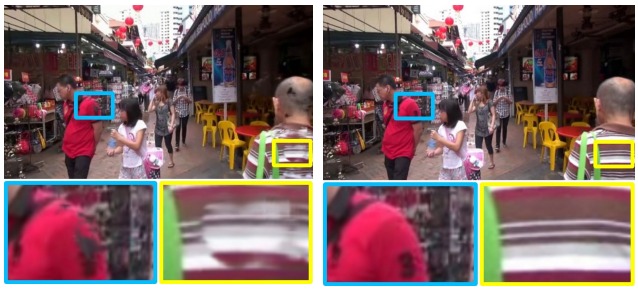

Figure 9. **Quilitative ablation of Color Correction.** The Color Correction module refining the projection enhances color accuracy, consequently reducing image artifacts.

Method	Distortion Value↑	PSNR↑
ENeRF	-	13.45
IBRNet	0.80	28.31
Full (Ours)	**0.90**	**40.01**
w/o Stabilized Rendering	0.87	23.56
w/o Adaptive Ray Range	0.81	37.83
w/o Color Correction	0.86	35.81

Table 2. **Quantitative results of ablation study**. We conduct comparative experiments of various range strategies and study the effect of each module. It should be noted that the results of ENeRF are so poor that the Distortion Value is unavailable.

Importance of Color Correction. We conduct a comparison between the results obtained by removing CC and using the full model. The presence of noticeable artifacts in the dynamic region in the left part of Fig. 9 leads to the decrease in PSNR comparing Row 6 and Row 3 of Table 1. This suggests that employing optical flow in CC to refine the projection can improve color accuracy.

6. Conclusions

In this paper, we propose a video stabilization framework termed RStab for integrating multi-frame fusion and 3D constraints to achieve full-frame generation and structure preservation. The core of RStab lies in **S**tabilized **R**endering, a volume rendering module utilizing both colors and features for multi-frame fusion in 3D space. To enhance **S**tabilized **R**endering module, we design an Adaptive Ray Range module for suppressing inconsistent information and a Color Correction module for refining color aggregation. By applying the three modules, RStab achieves full-frame generation with structure preservation and outperforms all previous stabilizers in FOV, image quality, and video stability across various datasets.

References

[1] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 5855–5864, 2021. 3

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 5470–5479, 2022. 3

[3] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 14124–14133, 2021. 3

[4] Yu-Ta Chen, Kuan-Wei Tseng, Yao-Chih Lee, Chun-Yu Chen, and Yi-Ping Hung. Pixstabnet: Fast multi-scale deep online video stabilization with pixel-based warping. In *Proceedings of IEEE International Conference on Image Processing (ICIP)*, pages 1929–1933, 2021. 2

[5] Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM Transactions on Graphics (TOG)*, 39(1):4:1–4:9, 2020. 1, 2, 6, 7

[6] Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, 31(5):126:1–126:10, 2012. 2

[7] Matthias Grundmann, Vivek Kwatra, and Irfan A. Essa. Auto-directed video stabilization with robust L1 optimal camera paths. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 225–232, 2011. 2, 6, 1

[8] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of International Conference on Learning Representations (ICLR)*, 2015. 6

[9] Yeong Jun Koh, Chulwoo Lee, and Chang-Su Kim. Video stabilization based on feature trajectory augmentation and selection and robust mesh grid warping. *IEEE Transactions on Image Processing (TIP)*, 24(12):5260–5273, 2015. 1, 2

[10] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung. Video stabilization using robust feature trajectories. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 1397–1404, 2009. 1, 2

[11] 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 IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 10621–10630, 2021. 1, 2, 4, 6, 7

[12] Chen Li, Li Song, Shuai Chen, Rong Xie, and Wenjun Zhang. Deep online video stabilization using IMU sensors. *IEEE Transactions on Multimedia (TMM)*, 25:2047–2060, 2023. 2

[13] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 6498–6508, 2021. 3

[14] Zhengqi Li, Qianqian Wang, Forrester Cole, Richard Tucker, and Noah Snavely. Dynibar: Neural dynamic image-based rendering. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 4273–4284, 2023. 3, 1

[15] Haotong Lin, Sida Peng, Zhen Xu, Yunzhi Yan, Qing Shuai, Hujun Bao, and Xiaowei Zhou. Efficient neural radiance fields for interactive free-viewpoint video. In *ACM SIGGRAPH Asia*, pages 39:1–39:9, 2022. 2, 3, 4, 8

[16] Kaimo Lin, Nianjuan Jiang, Shuaicheng Liu, Loong-Fah Cheong, Minh N. Do, and Jiangbo Lu. Direct photometric alignment by mesh deformation. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 2701–2709, 2017. 2

[17] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3d video stabilization. *ACM Transactions on Graphics (TOG)*, 28(3):44, 2009. 1, 2

[18] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4:1–4:10, 2011. 1, 2

[19] Shuaicheng Liu, Yinting Wang, Lu Yuan, Jiajun Bu, Ping Tan, and Jian Sun. Video stabilization with a depth camera. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 89–95, 2012. 1, 2

[20] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78:1–78:10, 2013. 1, 2, 6, 7

[21] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 4209–4216, 2014. 1, 2

[22] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 800–815, 2016. 2

[23] Shuaicheng Liu, Mingyu Li, Shuyuan Zhu, and Bing Zeng. Codingflow: Enable video coding for video stabilization. *IEEE Transactions on Image Processing (TIP)*, 26(7):3291–3302, 2017. 1, 2

[24] 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 IEEE International Conference on Computer Vision (ICCV)*, pages 2279–2288, 2021. 1, 2, 6, 7

[25] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 7210–7219, 2021. 3

[26] Andreas Meuleman, Yu-Lun Liu, Chen Gao, Jia-Bin Huang, Changil Kim, Min H. Kim, and Johannes Kopf. Progressively optimized local radiance fields for robust view synthesis. In *Proceedings of IEEE Conference on Computer Vision*

Pattern Recognition (CVPR), pages 16539–16548, 2023. 3, 1

[27] Ben Mildenhall, Pratul P. Srinivasan, Rodrigo Ortiz Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Transactions on Graphics (TOG)*, 38(4):29:1–29:14, 2019. 6

[28] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (TOG)*, 41(4):1–15, 2022. 3

[29] Simon Niklaus and Feng Liu. Softmax splatting for video frame interpolation. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 5436–5445, 2020. 5

[30] Zhenmei Shi, Fuhao Shi, Wei-Sheng Lai, Chia-Kai Liang, and Yingyu Liang. Deep online fused video stabilization. In *Proceedings of Winter Conference on Applications of Computer Vision (WACV)*, pages 865–873. IEEE, 2022. 2

[31] Brandon M. Smith, Li Zhang, Hailin Jin, and Aseem Agarwala. Light field video stabilization. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 341–348, 2009. 1, 2

[32] Alex Trevithick and Bo Yang. Grf: Learning a general radiance field for 3d representation and rendering. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 15182–15192, 2021. 3

[33] 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 Transactions on Image Processing (TIP)*, 28 (5):2283–2292, 2019. 6, 7, 1

[34] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P. Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas A. Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 4690–4699, 2021. 3, 4, 6, 8, 1

[35] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-nerf: Point-based neural radiance fields. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 5438–5448, 2022. 3

[36] Yufei Xu, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. DUT: learning video stabilization by simply watching unstable videos. *IEEE Transactions on Image Processing (TIP)*, 31:4306–4320, 2022. 2

[37] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 4578–4587, 2021. 3

[38] Jiyang Yu and Ravi Ramamoorthi. Selfie video stabilization. In *Proceedings of European Conference on Computer Vision (ECCV)*, pages 569–584, 2018. 2, 6, 7, 1

[39] Jiyang Yu and Ravi Ramamoorthi. Robust video stabilization by optimization in CNN weight space. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 3800–3808, 2019. 6

[40] Jiyang Yu and Ravi Ramamoorthi. Learning video stabilization using optical flow. In *Proceedings of IEEE Conference on Computer Vision Pattern Recognition (CVPR)*, pages 8156–8164, 2020. 2, 6, 7, 1

[41] Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE Transactions on Image Processing (TIP)*, 29:3582–3595, 2020. 2

[42] Weiyue Zhao, Xin Li, Zhan Peng, Xianrui Luo, Xinyi Ye, Hao Lu, and Zhiguo Cao. Fast full-frame video stabilization with iterative optimization. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, pages 23534–23544, 2023. 1, 2, 6, 7