# StabiGS: Video Stabilization through Rendering-Aware Trajectory Optimization in 3DGS-Reconstructed Scenes

Supplementary Material

## 1. Synthetic Dataset Overview

**Motivation** Evaluating video stabilization is challenging because there is a lack of ground-truth-stabilized videos to serve as a reference. Consequently, most evaluations are usually performed based on proxy metrics and subjective feedback to indicate perceptual quality. In pursuit of a more comprehensive evaluation, we propose a synthetic benchmark that leverages photorealistic 3D environments and rendering engines to generate both unstable and stable videos by rendering images given an input camera path within the scene. This setup provides full access to scene geometry and camera parameters, allowing us to generate reference sequences for any stabilization method by reproducing its stabilized camera trajectory.

**Videos and Scenes** We use Blender Software [2] (version 4.1.0) to generate six unstable sequences from five publicly available 3D scenes: Amazon Lumberyard Bistro [10], NVIDIA Emerald Square City [5], UE4 Sun Temple [4], Tropical Forest [13], and Scandinavian Decoration [3]. Each video contains 150 frames at full-HD resolution. A detailed description of the benchmark is provided in Tab. 1.

**Details** All sequences are rendered using Blender's Cycles engine (a physically-based path tracer), with GPU acceleration and denoising via NVIDIA OptiX. Each frame is generated with 512 samples per pixel to balance visual quality and rendering efficiency. Both unstable and ground-truth stabilized videos are rendered under identical scene configurations to ensure photometric consistency across comparisons. The shaky trajectories were generated by simulating realistic hand-held camera motion using a correlated AR(1) noise model applied to both translation and rotation, combined with periodic roll oscillations and vertical "bobbing" to reproduce gait-induced motion during walking. This produces camera paths that closely mimic natural human hand tremor and body motion while filming. We consider three motion types: forward walking, sideways walking, and intense rotation. Furthermore, we assume that all stabilization methods do not have access to ground-truth camera poses, as each method introduces its own bias during motion estimation (*e.g.*, PoseNet for RStab [11], Deep3D [7], and COLMAP [12] in our method). For fair comparison, we generated a ground-truth reference for each stabilization method by applying its respective smoothing strategy to the original trajectory. Thus, metrics $\mathbf{D}^*$ and $\mathbf{PSNR}^*$ are computed for each method between the estimated smooth video and its ground-truth reference. In particular, $\mathbf{D}^*$ quantifies the anisotropic scaling of the estimated homography between the ground-truth and output frames and $\mathbf{PSNR}^*$ evaluates the photometric fidelity as the average Peak Signal-to-Noise Ratio between ground-truth and output frames [1].

**Additional Results on Synthetic Videos** We provide more results on the synthetic benchmark in Fig. 1 and Tab. 2. We report averaged results across the synthetic videos for Deep3D [7], RStab [11], and two variants of our method: without rendering loss and with rendering loss.

Incorporating rendering loss slightly improves the photometric fidelity while maintaining stabilization performance. This indicates that rendering-aware optimization avoids pose drift without compromising trajectory smoothness. Qualitative results demonstrate that our method produces sharper rendering and fewer distortion artifacts than Deep3D [7] and RStab [11].

## 2. Implementation Details

**Rendering Loss Hyperparameter** We analyze the influence of the hyperparameter $\lambda$ on the Rendering Coverage Score ($\mathcal{RCS}$) in Fig. 2. Two viewpoints are considered: (a) a pose with a localized sparse region of Gaussians (poor coverage), indicating drift from well-reconstructed areas, and (b) a pose with dense coverage (good coverage). For each pose, we report $\mathcal{RCS}$ under varying $\lambda$ values. The design of $\mathcal{RCS}$ aims to differentiate (a) and (b) by emphasizing sparse regions through a softmin weighting controlled by $\lambda$. When $\lambda = 0$, $\mathcal{RCS}$ reduces to a uniform mean over opacities, making the score less sensitive to small sparse regions. Hence, such poses are not strongly penalized in $\mathcal{L}_{\text{render}}$. Increasing $\lambda$ applies a *softmin* weighting that emphasizes low-opacity pixels, allowing $\mathcal{RCS}$ to better capture and penalize localized coverage gaps due to pose drift.

**Computation Time** Tab. 3 reports the estimated computation time for each component of our framework on the synthetic *temple* video. All timings were measured on a single NVIDIA A100 GPU.

[1] The star ($^*$) in the notation $\mathbf{D}^*$ and $\mathbf{PSNR}^*$ indicates that the metric is calculated using the ground truth reference in the case of synthetic dataset.

| Video | 3D Scene | Motion Type | Frames | Resolution | FPS |
|---|---|---|---|---|---|
| bistro | Amazon Lumberyard Bistro [10] | forward walking | 150 | 1920×1080 | 30 |
| forest | Tropical Forest [13] | sideways walking | 150 | 1920×1080 | 30 |
| house | Scandinavian Decoration [3] | sideways walking | 150 | 1920×1080 | 30 |
| square | NVIDIA Emerald Square City [5] | intense rotation | 150 | 1920×1080 | 30 |
| street | Amazon Lumberyard Bistro [10] | forward walking | 150 | 1920×1080 | 30 |
| temple | UE4 Sun Temple [4] | intense rotation | 150 | 1920×1080 | 30 |

Table 1. **Overview of the synthetic benchmark** comprising six unstable sequences rendered from diverse photorealistic 3D environments.

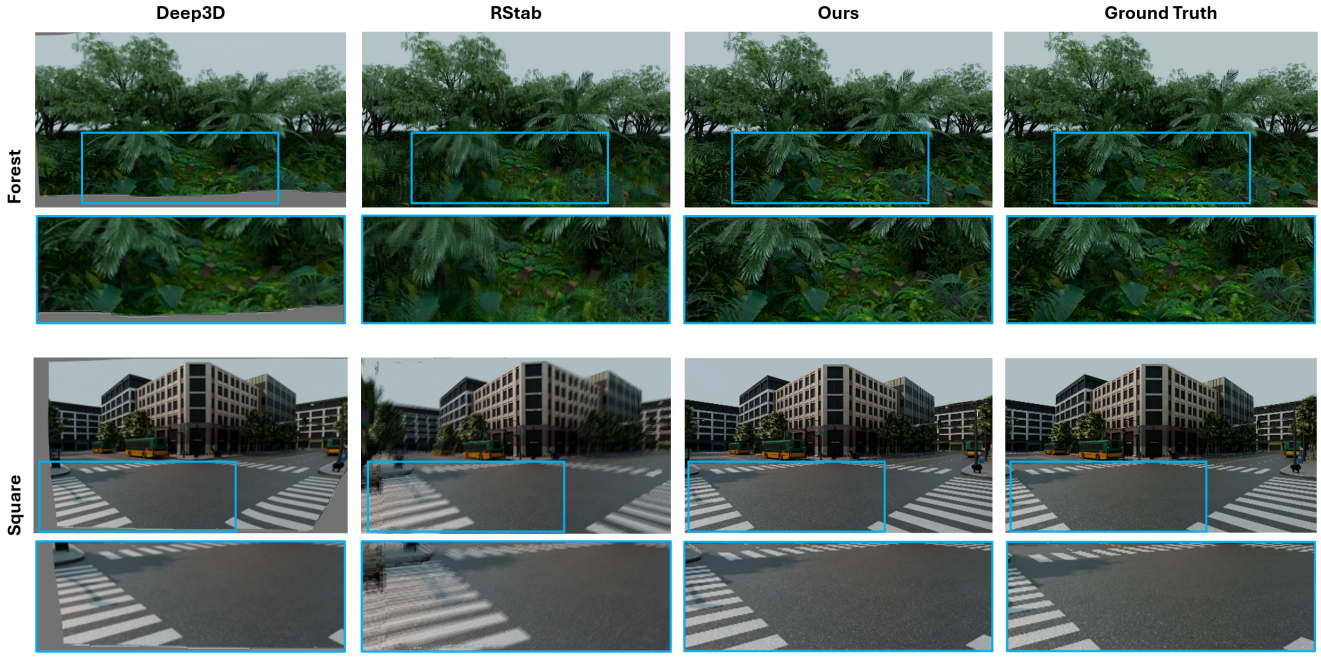


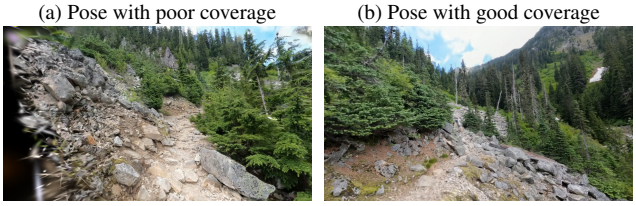

Figure 1. **Qualitative results from *forest* and *square* synthetic scenes** comparing Deep3D [7], RStab [11], our method, and the ground-truth rendering. **Cyan** boxes highlight regions with cropping artifacts and distortions, which are further shown in the zoomed-in views.

| Method | *Stabilization metrics* | | | *Reference-based metrics* | |
|---|---|---|---|---|---|
| | D↑ | S↑ | ITF↑ | D*↑ | PSNR*↑ |
| Deep3D [7] | 0.93 | 0.86 | 33.56 | 0.88 | 22.00 |
| RStab [11] | 0.96 | 0.88 | 32.21 | 0.85 | 22.63 |
| Ours (w/o Rendering Loss) | 0.75 | 0.93 | 33.17 | 0.99 | 24.58 |
| Ours | 0.71 | 0.93 | 32.79 | 0.99 | 24.64 |

Table 2. **Additional results on synthetic videos.** We report stabilization and reference-based metrics averaged over synthetic videos.

**Additional Qualitative Results** We provide additional comparisons versus 2D methods on intense camera motion in Fig. 3.

(a) Pose with poor coverage (b) Pose with good coverage



| Pose Type | $\lambda = 0$ | $\lambda = 1$ | $\lambda = 5$ |
|---|---|---|---|
| Poor coverage | 0.87 | 0.78 | 0.22 |
| Good Coverage | 0.96 | 0.95 | 0.85 |

Figure 2. **Effect of $\lambda$ on Rendering Coverage Score ($\mathcal{RCS}$):** we illustrate the impact of $\lambda$ for two poses: one with dense coverage (good coverage) and one containing a localized sparse region (poor coverage). For each pose, we compute $\mathcal{RCS}$ under varying $\lambda$ values.

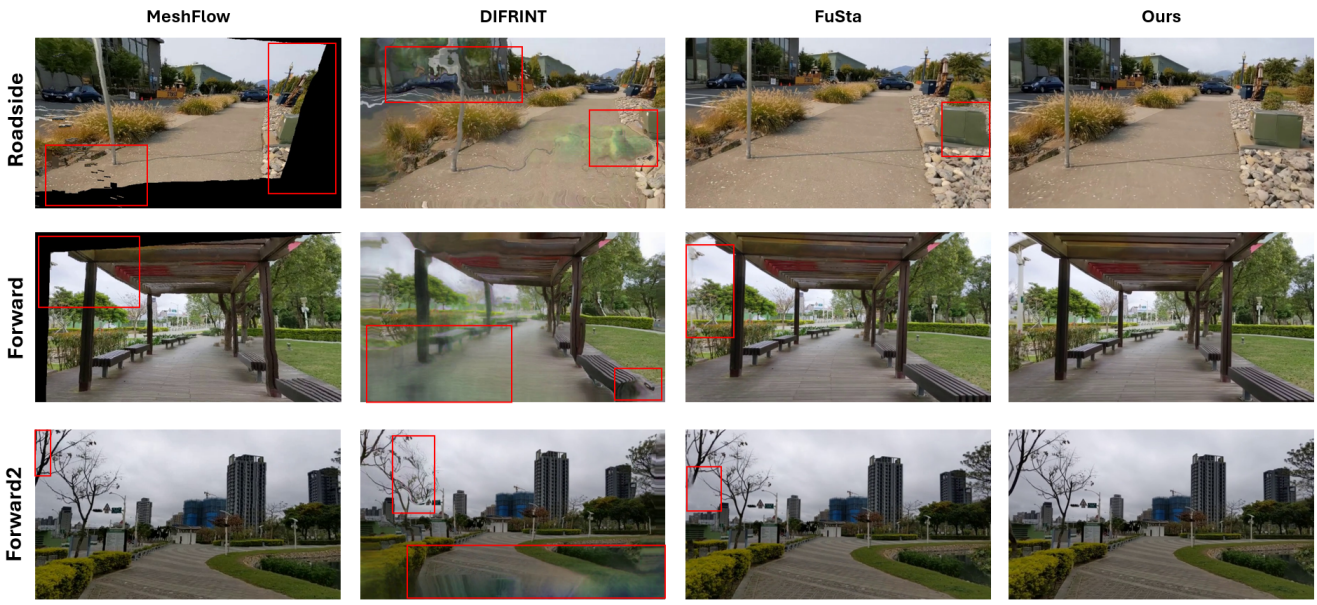



Figure 3. **Qualitative results against 2D baselines on *Intense* videos.** The **red** boxes indicate rendering artifacts. We compare our method with Meshflow [8] , DIFRINT [1] and FuSta [9]. DIFRINT faces fatal failure in the case of intense camera motion. MeshFlow is prone to cropping and geometric deformation. FuSta suffers from distortion and shear artefacts, especially near frame boundaries. Our method consistently produces sharp, stabilized frames while preserving scene consistency.

| Task | Time |
|---|---|
| Pose Estimation and Sparse (COLMAP) | 23 min |
| 3DGS Reconstruction | 24 min |
| Trajectory Optimization | 10 min |
| Rendering | $\leq$ 1 min |

Table 3. Average runtime of each component for synthetic *temple* video (150 Full-HD frames).

## 3. User Study

Our user study involved **70 participants**, divided into two groups according to their familiarity with video stabilization: 45 experts and 25 non-experts. We conducted the study via an online Microsoft Form, with an example question shown in Fig. 4. The form included 10 videos of static scenes, with side-by-side, synchronized, stabilized videos produced by our method and GaVS [14]. The mapping of methods to videos was randomized to ensure that (A) and (B) did not consistently correspond to the same method. For each pair, participants were asked to select the video they perceived as more stable, with the option to choose a comparable performance. Participants could replay and pause the videos multiple times. The detailed voting results are shown in Fig. 5. Both groups strongly favored our method, especially for intense videos. Mild videos show slightly more comparable votes, suggesting that differences

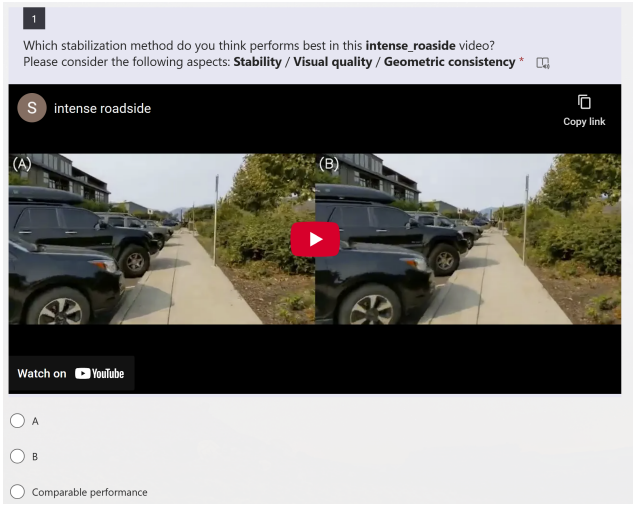
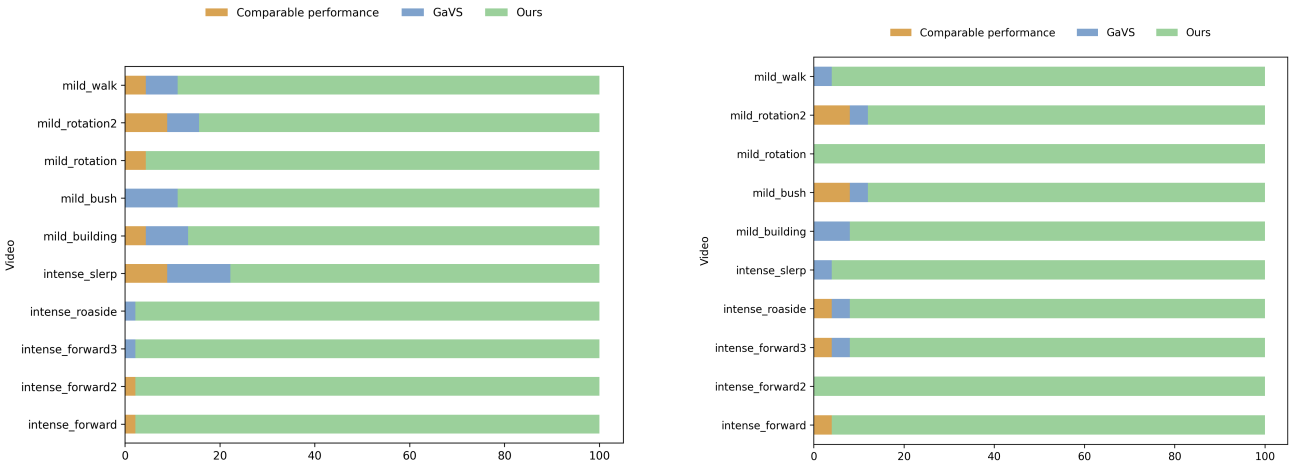



Figure 4. User Study Example Question

are less noticeable for low-intensity camera motion.

## 4. Dynamic Objects Case

Our approach builds upon 3D Gaussian Splatting [6], which inherently assumes a static scene geometry. Consequently, dynamic objects are either poorly reconstructed or removed during the optimization process. We illustrate this behavior in Fig. 6.




(a) Experts (b) Non-Experts

Figure 5. **Detailed User study votes** from 45 vision experts and 25 non-expert viewers across all *Mild* and *Intense* videos.


Figure 6. **Dynamic Object Case:** dynamic elements such as vehicles (red boxes) are either poorly reconstructed or eliminated due to the static-scene assumption.

## References


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

[2] Blender Online Community. Blender - a 3d modelling and rendering package. http://www.blender.org, 2018. 1

[3] Priscila De Melo. blender scene: Scandinavian decoration. https://www.blenderkit.com/asset-gallery-detail/4d2355c2-a20b-4be5-a1ba-3b0217b6730e/, 2025. [Online; accessed 11-Nov-2025]. 1, 2

[4] Epic Games. Unreal engine sun temple, open research content archive (orca). https://developer.nvidia.com/ue4-sun-temple, 2017. [Online; accessed 11-Nov-2025]. 1, 2

[5] Nicholas Hull, Kate Anderson, and Nir Benty. Nvidia emerald square, open research content archive (orca). https://developer.nvidia.com/orca/nvidia-emerald-square, 2017. [Online; accessed 11-Nov-2025]. 1, 2

[6] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (TOG)*, 42(4), 2023. 3

[7] Yao-Chih Lee, Kuan-Wei Tseng, Yu-Ta Chen, Chien-Cheng Chen, Chu-Song Chen, and Yi-Ping Hung. 3d video sta-

bilization with depth estimation by cnn-based optimization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10616–10625, 2021. 1, 2

[8] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision (ECCV)*, pages 800–815, Cham, 2016. 3

[9] 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 *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2279–2288, 2021. 3

[10] Amazon Lumberyard. Amazon lumberyard bistro, open research content archive (orca). http://developer.nvidia.com/orca/amazon-lumberyard-bistro, 2017. [Online; accessed 11-Nov-2025]. 1, 2

[11] Zhan Peng, Xinyi Ye, Weiyue Zhao, Tianqi Liu, Huiqiang Sun, Baopu Li, and Zhiguo Cao. 3d multi-frame fusion for video stabilization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7507–7516, 2024. 1, 2

[12] Johannes L. Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4104–4113, 2016. 1

[13] Peeraphon Viriyahirunpaiboon. blender scene: Tropical forest. https://www.blenderkit.com/asset-gallery-detail/2ac89235-24d6-4786-a0df-aead0090ae3a/, 2025. [Online; accessed 11-Nov-2025]. 1, 2

[14] Zinuo You, Stamatios Georgoulis, Anpei Chen, Siyu Tang, and Dengxin Dai. Gavs: 3d-grounded video stabilization via temporally-consistent local reconstruction and rendering. In *Special Interest Group on Computer GRAPHics and Interactive Techniques (SIGGRAPH)*, New York, NY, USA, 2025. 3