Video Stitching from Multiple Moving Cameras

Shira Ifergane **Roy Amoyal** **Shahar Benishay** **Oren Freifeld**

The Stein Faculty of Computer and Information Science,
Ben-Gurion University of the Negev, Israel

{shiraif,amoyalr,shbeni}@post.bgu.ac.il, orenfr@bgu.ac.il

Abstract

We address geometric video alignment for the same scene captured by multiple freely moving cameras. Existing methods, which are based on image stitching or video stabilization, are typically limited to two cameras or must assume the restrictive setting of a fixed linear ordering of the cameras. We propose MViS, a lightweight test-time optimization framework for joint multi-view alignment with temporal consistency. MViS uses a spatial-transformer-inspired net to jointly process frames from all views within a short temporal window and predict per-view, per-time homographies. Optimization proceeds in two stages: geometry-guided alignment followed by appearance-based refinement, with both stages enforcing multi-view and temporal consistency across the sequence. The method requires no scene-specific hyperparameter tuning, handles arbitrary and time-varying view-intersection graphs, and remains robust under rapid camera motion. On a standard two-view video benchmark, MViS matches or surpasses the state of the art. To facilitate evaluation in more complex scenarios, we introduce an annotated dataset with more than two freely moving cameras and establish the first benchmark for this setting. Project page: https://bgu-cs-vil.github.io/MViS-project/.

1 Introduction

Geometric alignment of multiple videos of the same scene captured by freely moving cameras is a challenging and underexplored task arising in multi-user handheld capture, drone footage, or mobile multi-camera devices. The goal is to estimate, for every frame in every view, transformations that bring all videos into a spatially aligned and temporally coherent configuration.

Applying Multiple Image Stitching (MIS) independently at each time yields spatially coherent but temporally inconsistent panoramas. The resulting video exhibits spatially dependent shaking that standard stabilization cannot correct, since different regions require conflicting temporal corrections that themselves may vary from frame to frame. This calls for a video-specific formulation that enforces spatial and temporal consistency jointly. Although pairwise alignment of moving-camera videos has been studied, extending it beyond two cameras is problematic: without global consistency, errors compound across views and time. Moreover, camera ordering and view intersections may vary over time, violating assumptions of fixed layouts or persistent overlaps.

Video stitching has long been treated as a hybrid of image stitching and video stabilization, either handling the two tasks sequentially in constrained setups or addressing them jointly. Most moving-camera methods still follow these pipelines and are limited to two cameras.

A prime example is StabStitch [1], which handles two moving cameras, and StabStitch++ [2], which extends to three cameras but only under a fixed linear spatial ordering. No existing method supports more than two freely moving cameras, where the view-intersection graph (defined by pairwise field-of-view overlaps) is arbitrary or time-varying. More broadly, existing approaches operate pairwise and lack a principled mechanism for global consistency across views and time.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

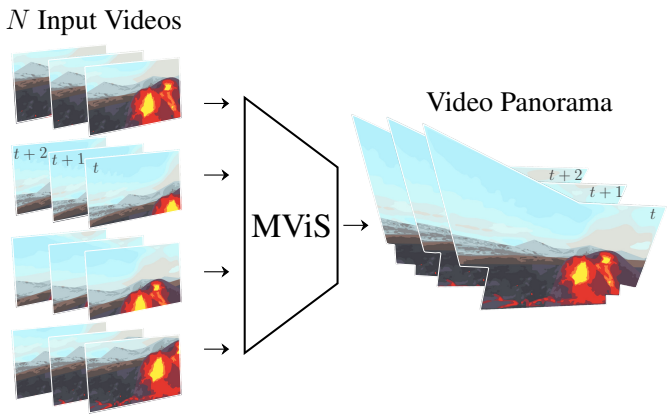


Figure 1: MViS aligns N moving-camera videos, without assuming a known/fixed topology of the view-intersection graph and requires no scene-specific hyperparameter tuning.

We propose Multiple Video Stitching (MViS), an unsupervised optimization method that aligns videos from multiple freely moving cameras (see Fig. 1) without requiring fixed or known camera ordering. It only assumes that the view-intersection graph is connected at each time. MViS uses a lightweight alignment network (inspired by, but different from, the Spatial Transformer Network (STN) [3]) that takes a local multi-view temporal window as input and jointly predicts the corresponding per-frame transformations. These predictions are optimized using losses that enforce multi-view and temporal consistency through relative compositions of transformations (*i.e.*, composing one per-frame transformation with the inverse of another) for pairs of frames, either across views at the same time or across consecutive times in the same view. This yields a scalable optimization that enforces global consistency over cameras and time, rather than independent pairwise alignments.

The optimization proceeds in two stages: geometry-guided alignment based on noisy pairwise transformations, followed by appearance-based refinement. The resulting regularization-free losses enable stable stitching under fast motion and time-varying view-intersection graphs without per-scene hyperparameter tuning. Operating on short time windows allows MViS to scale to hundreds or thousands of frames while adapting to diverse capture conditions.

Our key contributions are:

1) A method for stitching videos from multiple freely moving cameras, the first in this setting to handle more than two cameras with arbitrary, time-varying view-intersection graphs.

2) A scalable unsupervised formulation that jointly aligns all views over time, while avoiding per-scene hyperparameter tuning through regularization-free losses.

3) A new annotated multi-view video dataset and benchmark for video stitching under unconstrained multi-camera motion.

4) A variance-based geometric metric for assessing alignment quality in multi-view video stitching.

2 Related Work

Multiple Image Stitching (MIS). Classical MIS methods often follow an incremental pairwise strategy, stitching one image at a time in a predefined order, which makes them prone to error accumulation and drift. A popular alternative, AutoStitch [4], addresses this by jointly optimizing the transformations of all images. Yet, despite many MIS methods (*e.g.*, [5, 6, 7, 8]), most classical and learning-based methods still handle only two input images. Methods supporting more than two images often rely on predefined sequences or spatial layout assumptions, such as known left-right ordering or a fixed reference image [9, 10]. In multiple video stitching from freely moving cameras, MIS methods produce good panoramas at individual time steps but yield temporally incoherent results when applied frame by frame. This failure, along with the difficulty of post-hoc video stabilization, stems from the fact that, due to independent camera motions, different panoramic regions usually move differently between times t and $t+1$, which per-frame stitching cannot capture.

Video Stitching. Early works handled the task sequentially, first stitching and then stabilizing videos [11], but only in restricted setups with stationary cameras or fixed rigs. Lin *et al*. [12] instead stabilized videos before stitching, but showed results only for two or three cameras (and, to date, have not released code). Several methods [13, 14, 15] incorporate video stabilization techniques such as bundled camera paths [16]. Among these, Guo *et al*. [13] first jointly optimized stitching and stabilization for moving cameras. Although they suggested extending the method beyond two cameras, it relied on camera-by-camera optimization with an alternating bundled-path scheme, resulting in high computational cost and limited scalability. Subsequent works (*e.g*., [14]) noted that such optimization is prohibitively hard. Moreover, the authors of [13] released no code, and the multi-camera extension was not demonstrated experimentally. In contrast, MViS is simple and scalable: each stage optimizes its loss over all views and sliding windows with standard solvers (*e.g*., ADAM), without alternating procedures or hand-crafted schedules. Perazzi *et al*. [17] stitch videos from linearly arranged wide-baseline cameras, seemingly using calibrated camera parameters. Similarly, [18] stitch videos from fixed camera rigs. The restrictive assumptions of both methods make them incompatible with unconstrained setups of multiple freely moving cameras, as also noted in [1].

The methods above improve short-term coherence, but remain limited to two-camera setups, fixed arrays, or stationary capture, and are inapplicable to more general, practical settings. Other methods target constrained capture setups. Lai *et al*. [19] propose a learning-based approach for linear camera arrays, training a pushbroom-style interpolation network that learns dense flow between adjacent views. Lo *et al*. [20] focus on dual-fisheye 360° cameras with geometric calibration and seam optimization. Both assume rigid, synchronized hardware and do not extend to freely moving cameras. The state-of-the-art (SOTA) video stitching method, StabStitch [1], is the most relevant to our work. It stitches two videos and jointly performs stabilization in an unsupervised, online framework based on UDIS++ [21]. Its slower extension, StabStitch++ [2], improves alignment and stabilization via a differentiable bidirectional decomposition and a warp-smoothing model. StabStitch++ can handle three cameras, but only with a fixed linear ordering, and does not generalize to unknown, dynamic topology or beyond three videos. In contrast, MViS handles arbitrary numbers of cameras and flexible view-intersection graphs.

Image Alignment. Several methods [22, 23, 24] jointly align tens of still images using high-dimensional DINO features [25]. They act on still images with masked objects and do not scale to videos: even [24], the most efficient, runs out of memory when the number of frames exceeds 70. While MViS shares with [24] the idea of comparing images via relative transformations, its formulation, architecture, and losses differ: MViS acts on videos and processes multi-view temporal windows rather than single images. Moreover, [24] is quadratic in the number of images, while MViS is linear in the number of frames and quadratic only in the (typically small) number of cameras.

Keypoint matchers such as RoMa [26] or LoFTR [27] operate on image pairs and do not account for temporal or multi-view consistency. MViS uses such matchers solely to generate noisy observed relative transformations that guide the geometric stage.

3 Method

Consider N approximately synchronized videos, $(V_i)_{i=1}^{N}$, of the same scene, captured by moving cameras. Each V_i has τ frames, $(F_i^t)_{t=1}^{\tau}$. We do not assume that the two views in every pair overlap at time t, only that the view-intersection graph, defined by field-of-view overlaps, is connected at each t. Our task is to estimate, for each F_i^t, a homography $\boldsymbol{H}_i^t$ that aligns it to a single panoramic coordinate system shared across all views and times. We now describe MViS, outlined in Fig. 2. After preprocessing (§ 3.1), which includes the extraction of noisy pairwise homographies, we use an alignment net (§ 3.2) to optimize view- and time-specific transformations in two stages: geometric-loss minimization (§ 3.3) followed by appearance-based refinement (§ 3.4). Both losses are defined using absolute per-view, per-time transformations but depend only on relative compositions, making them invariant to a global transformation and regularization-free, as explained in § 3.5. Finally, we generate a panoramic video as described in § 3.6, while § 3.7 explains the implicit temporal stabilization.

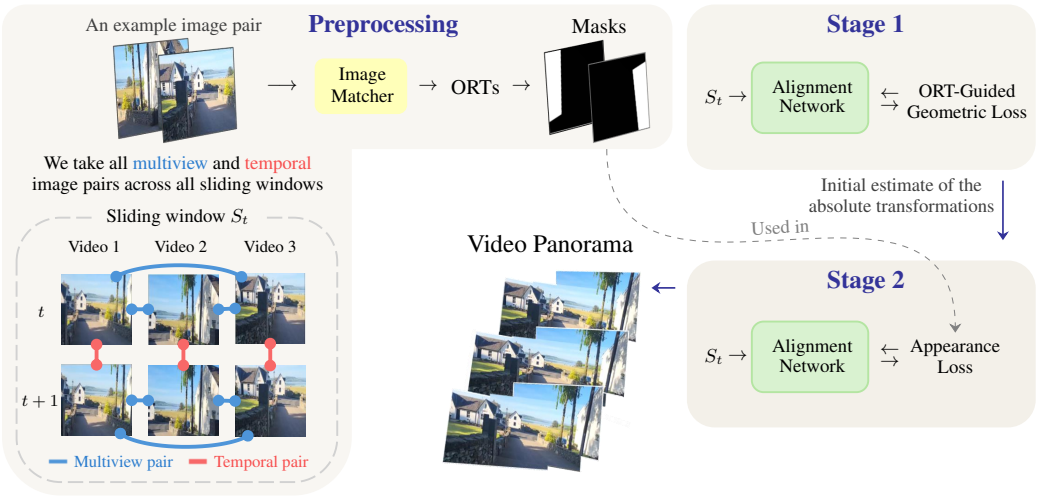


Figure 2: An overview of the proposed method, Multiple Video Stitching (MViS).

3.1 Preprocessing

The preprocessing has four steps: resizing, RGB normalization, Observed Relative Transformation (ORT) estimation, and mask extraction. We first **resize** each frame F_i^t to a fixed square resolution $r \times r$ ($r = 256$ in all experiments), making runtime independent of the original resolution. We then **normalize** each RGB channel by subtracting its sample mean and dividing by its sample standard deviation, computed across all frames in all videos. This simple normalization handles inter-view color variations effectively. Let $I_i^t \in \mathbb{R}^{r \times r \times 3}$ denote the resulting image. Next, we estimate noisy pairwise homographies, which we call **Observed Relative Transformations (ORTs)**, for view pairs (I_i^t, I_j^t), $i \neq j$, and temporal pairs (I_i^t, I_i^{t+1}). For each such pair, we extract keypoint correspondences using an image matcher and then use RANSAC to estimate the homography. Let $\widehat{\boldsymbol{H}}_{i\to j}^t$ denote the ORT from I_i^t to I_j^t, let $\widehat{\boldsymbol{H}}_i^{t\to t+1}$ denote the ORT from I_i^t to I_i^{t+1}, and let $\widehat{\boldsymbol{H}}_i^{t+1\to t}$ denote the ORT from I_i^{t+1} to I_i^t. Due to estimation noise, typically $\widehat{\boldsymbol{H}}_i^{t+1\to t} \neq (\widehat{\boldsymbol{H}}_i^{t\to t+1})^{-1}$.

The ORTs and the overlap masks they define affect the losses below. At a given t, I_i^t and I_j^t may have an empty or negligible spatial intersection. We detect such cases using matcher-provided certainty scores: if the average score for an image pair is below a threshold (*e.g.*, 0.3 for RoMa), the losses ignore that pair; otherwise, we assume a nonempty intersection. Accordingly, let $M_{i\to j}^t$ be the **binary mask** obtained by warping an r-by-r all-ones mask from the domain of I_i^t to that of I_j^t using $\widehat{\boldsymbol{H}}_{i\to j}^t$. This mask represents the overlapping image region between I_j^t and the warped I_i^t. Similarly, $M_i^{t\to t+1}$ and $M_i^{t+1\to t}$ are defined via $\widehat{\boldsymbol{H}}_i^{t\to t+1}$ and $\widehat{\boldsymbol{H}}_i^{t+1\to t}$.

3.2 The Architecture of the Alignment Network

MViS uses an alignment network (Fig. 3) inspired by the Spatial-Transformer Net (STN) [3], but adapted to multi-view video alignment. In a standard STN, the input is a single image and it predicts a single transformation. In contrast, here the net's input is a sliding window $(I_1^t, \ldots, I_N^t, I_1^{t+1}, \ldots, I_N^{t+1})$, and it predicts the parameters of $2N$ homographies, $(\boldsymbol{\theta}_1^t, \ldots, \boldsymbol{\theta}_N^t, \boldsymbol{\theta}_1^{t+1}, \ldots, \boldsymbol{\theta}_N^{t+1})$. A shared lightweight CNN first processes each image independently. A fully-connected module then aggregates information across the window and jointly predicts, for each frame in the window, an 8-dimensional vector $\boldsymbol{\theta}_i^t$ parameterizing a homography $\boldsymbol{H}_i^t = \mathrm{func}(\boldsymbol{\theta}_i^t)$, a 3-by-3 invertible matrix with determinant 1. That well-known parameterization (used, *e.g.*, in [28, 29, 24, 30]) is based on the Special Linear group $\mathrm{SL}(3)$ and detailed in Appendix B. It guarantees matrix invertibility; this invertibility improves optimization stability [31] and is key to our relative-composition losses below. The fully-connected module has two linear layers separated by a GELU activation [32]. The full architecture appears in Appendix D. We note two differences from existing STN-based joint-alignment methods for images (*e.g.*, [22, 24]). 1) These methods process

one image at a time, so images interact only through the loss. In contrast, MViS processes all $2N$ images in the sliding window jointly, allowing cross-view and temporal interactions already inside the net. 2) While STN [3] has an image-warping module, MViS uses image warping only in the second-stage appearance loss; the geometric loss acts on sparse grid points so it requires no image warping.

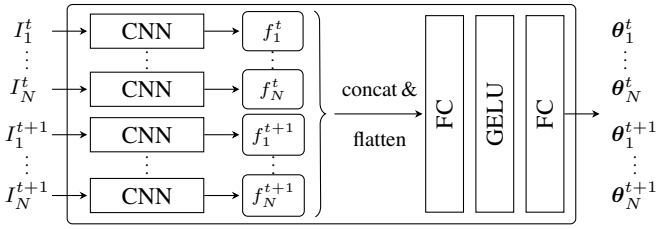


Figure 3: The alignment network for N cameras. The weights of the CNNs are shared.

3.3 Stage 1: The ORT-guided Geometric Loss

Minimizing the appearance-based loss in the next section (§ 3.4) is highly effective, but typically requires a good initialization. We therefore first optimize a geometric loss (depicted in Fig. 4) guided by the ORTs from § 3.1. Unlike the appearance loss, this loss yields reasonable results even when the initial transformations are set to the identity. Let $\boldsymbol{G} \in \mathbb{R}^{d^2 \times 2}$ contain the d^2 2D points of a regular $d \times d$ grid over $[-1, 1] \times [-1, 1]$, with the image center as the origin. In our experiments, $d = 7$. For a 3-by-3 homography matrix $\boldsymbol{H}$, we slightly abuse notation and let $\boldsymbol{H}(\boldsymbol{G}) \in \mathbb{R}^{d^2 \times 2}$ denote the result, in inhomogeneous coordinates, of applying $\boldsymbol{H}$ to $\boldsymbol{G}$.

Our geometric loss, for sliding window $S_t = \{I_i^s \mid s \in \{t, t+1\}, i \in \{1, \ldots, N\}\}$,

is $\mathcal{L}_{\text{geo}} = \mathcal{L}_{\text{geo}}^{\text{time}} + \mathcal{L}_{\text{geo}}^{\text{views}}$ where these terms are

$$\mathcal{L}_{\text{geo}}^{\text{views}} = \frac{1}{2(N^2-N)} \sum_{i \neq j} \sum_{k \in \{0,1\}} \mathcal{L}^{\text{smooth-}\ell_1} \Big(\underbrace{\widehat{\boldsymbol{H}}_{i \to j}^{t+k}}_{\text{ORT}}(\boldsymbol{G}) - ((\boldsymbol{H}_j^{t+k})^{-1} \boldsymbol{H}_i^{t+k})(\boldsymbol{G}) \Big) \tag{1}$$

$$\mathcal{L}_{\text{geo}}^{\text{time}} = \mathcal{L}_{\text{geo}}^{\text{fwd}} + \mathcal{L}_{\text{geo}}^{\text{bwd}}\,, \tag{2}$$

$$\mathcal{L}_{\text{geo}}^{\text{fwd}} = \frac{1}{2N} \sum_{i=1}^{N} \mathcal{L}^{\text{smooth-}\ell_1} \Big(\underbrace{\widehat{\boldsymbol{H}}_i^{t \to t+1}}_{\text{ORT}}(\boldsymbol{G}) - ((\boldsymbol{H}_i^{t+1})^{-1} \boldsymbol{H}_i^t)(\boldsymbol{G}) \Big), \tag{3}$$

$$\mathcal{L}_{\text{geo}}^{\text{bwd}} = \frac{1}{2N} \sum_{i=1}^{N} \mathcal{L}^{\text{smooth-}\ell_1} \Big(\underbrace{\widehat{\boldsymbol{H}}_i^{t+1 \to t}}_{\text{ORT}}(\boldsymbol{G}) - ((\boldsymbol{H}_i^{t})^{-1} \boldsymbol{H}_i^{t+1})(\boldsymbol{G}) \Big) \tag{4}$$

and $\mathcal{L}^{\text{smooth-}\ell_1}$ is the smooth ℓ_1 loss. For each image pair, $\mathcal{L}_{\text{geo}}$ compares two transformations of $\boldsymbol{G}$: 1) the ORT and 2) the composition of the predicted transformation for one image with the inverse of

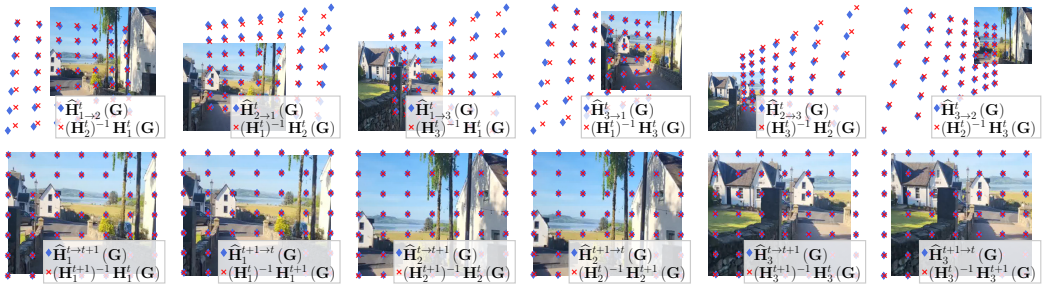


Figure 4: **Stage 1's ORT-guided Geometric Loss**, $\mathcal{L}_{\text{geo}}$. Illustration using a sliding window S_t with $N = 3$. Top row: the inter-view pairs. Bottom: the temporal pairs. In each plot, the grid warped by the predicted transformation is overlaid on the grid warped by the ORT estimate. Each image corresponds to the target frame of the transformation (*e.g.*, top-left: $I_1^t \to I_2^t$, visualized over I_2^t).

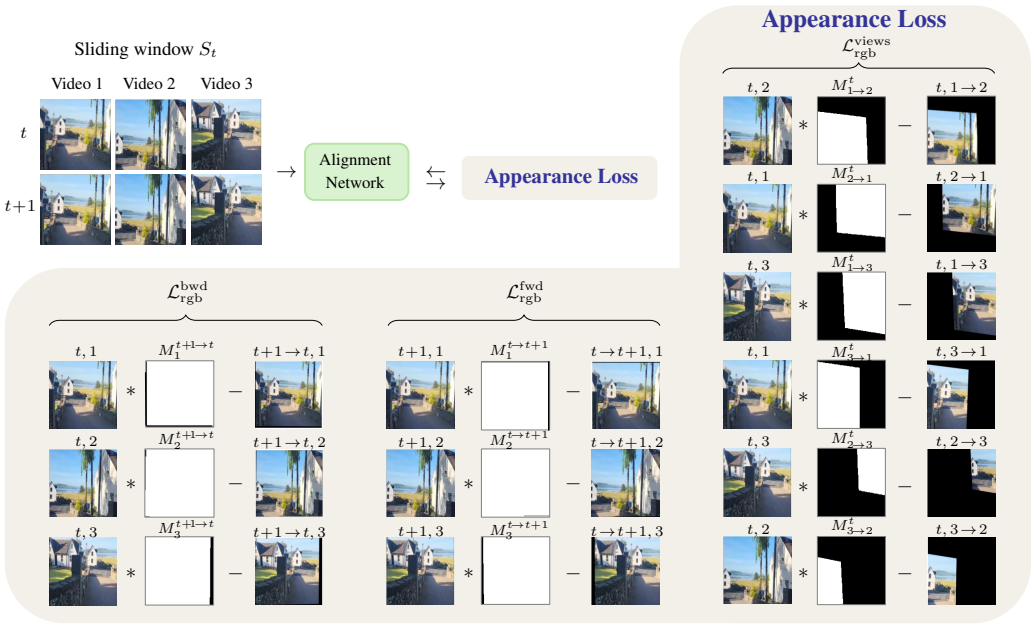


Figure 5: **Stage 2's Appearance Loss $\mathcal{L}_{\text{rgb}}$**. In this figure, $N = 3$.

the transformation predicted for the other. During optimization, the overall geometric loss is averaged over all sliding windows in a batch.

3.4 Stage 2: The Appearance Loss

The second stage uses an appearance loss based on (normalized) RGB values rather than transformed grid points. In another slight abuse of notation, for a homography matrix $\boldsymbol{H}$ and image I, let $\boldsymbol{H}(I)$ denote the result of warping I by $\boldsymbol{H}$. For each sliding window S_t, the proposed appearance-based loss is $\mathcal{L}_{\text{rgb}} = \mathcal{L}_{\text{rgb}}^{\text{time}} + \mathcal{L}_{\text{rgb}}^{\text{views}}$, where

$$\mathcal{L}_{\text{rgb}}^{\text{views}} = \frac{1}{2(N^2-N)} \sum_{i \neq j} \sum_{k \in \{0,1\}} ||I_j^{t+k} - ((\boldsymbol{H}_j^{t+k})^{-1} \boldsymbol{H}_i^{t+k})(I_i^{t+k})||_{\ell_2}^2 , \tag{5}$$

$$\mathcal{L}_{\text{rgb}}^{\text{time}} = \mathcal{L}_{\text{rgb}}^{\text{fwd}} + \mathcal{L}_{\text{rgb}}^{\text{bwd}} , \tag{6}$$

$$\mathcal{L}_{\text{rgb}}^{\text{fwd}} = \frac{1}{2N} \sum_{i=1}^{N} ||I_i^{t+1} - ((\boldsymbol{H}_i^{t+1})^{-1} \boldsymbol{H}_i^{t})(I_i^{t})||_{\ell_2}^2 , \tag{7}$$

$$\mathcal{L}_{\text{rgb}}^{\text{bwd}} = \frac{1}{2N} \sum_{i=1}^{N} ||I_i^{t} - ((\boldsymbol{H}_i^{t})^{-1} \boldsymbol{H}_i^{t+1})(I_i^{t+1})||_{\ell_2}^2 , \tag{8}$$

and $||\cdot||_{\ell_2}$ is the ℓ_2 norm. As in § 3.3, we average the losses across all sliding windows in a batch. In each summand of Eq. 5, Eq. 7, and Eq. 8, the image difference is computed only in the overlap region determined by the appropriate mask, which in turn is induced by the relevant ORT (see § 3.1). We omit the masks from the notation to avoid clutter; Fig. 5 depicts the loss and the masked regions.

3.5 Regularization-free Losses and Global Invariance

The losses from § 3.3 and § 3.4 are defined via compositions of predicted absolute transformations; *e.g.*, for views i and j, a loss term depends on the relative transformation $\boldsymbol{H}_j^{-1}\boldsymbol{H}_i$. In the geometric stage, this transformation is applied to corresponding points; in the appearance stage, to image content. A central consequence is that no explicit regularization over the predicted transformations is required. Each transformation must explain observed data, either geometric correspondences or image appearance. Degenerate transformations (*e.g.*, excessive shrinking or severe distortion) would increase inter-view misalignment and hence the loss. Thus, the loss cannot be trivially reduced through collapse or uncontrolled growth of the transformations. Consequently, regularization terms (*e.g.*, on transformation size or lack of smoothness) are unneeded, freeing us from scene-specific

hyperparameter tuning and the optimization complexity such regularization entails (see also § C.2). Note that regularization-free losses and their associated benefits have also been demonstrated in several of our earlier joint-alignment works in various other settings, including [33, 24, 30, 34].

A second property is that no explicit atlas or canonical panorama is needed: all comparisons are made directly between views rather than to a learned template. This is important in multi-camera video, where the scene evolves over time and maintaining a stable global reference is nontrivial. Crucially, although the loss terms are defined over pairs of views, MViS jointly predicts the transformations of all views in the sliding window, yielding a globally consistent alignment rather than a collection of independent pairwise alignments.

Lastly, the losses are invariant to a common global transformation: if all predicted transformations are left-composed with some transformation $\boldsymbol{G}$, then their relative transformations are unchanged, since $(\boldsymbol{G}\boldsymbol{H}_j)^{-1}(\boldsymbol{G}\boldsymbol{H}_i) = \boldsymbol{H}_j^{-1}\boldsymbol{H}_i$. Thus, the minimizers are defined only up to a global transformation and form an equivalence class under such compositions. While this implies infinitely many solutions in terms of *transformations* (a phenomenon shared with [24, 30] and works on synchronization over Lie groups, *e.g.*, [35, 29], or SfM), all members of a given equivalence class induce the same inter-frame *correspondences*. In practice, for visualization, we apply to all predicted transformations the inverse of the predicted transformation of an arbitrarily-chosen frame (*e.g.*, the first frame of the first video). Since this choice is post-hoc, it has no effect on alignment quality. Finally, while relative-transformation losses were recently used in [24, 30], their roots go back to classical methods such as [36] and even [37].

3.6 From the Prediction to a Panoramic Video

After optimization, we create a panoramic video by applying the predicted $\boldsymbol{H}_i^t$ to the frames F_i^t, while accounting for the resizing. Since the solution is defined up to a global transformation, visualization requires choosing a reference. We either apply, at each t, the inverse of $\boldsymbol{H}_1^t$ to all frames at that time, or apply the inverse of $\boldsymbol{H}_1^1$ to all frames across all times and views. This postprocessing is purely for visualization: MViS gives no special weight to the first video or to its first frame.

3.7 Temporal Stabilization

MViS implicitly obtains temporal stabilization through three mechanisms: 1) overlapping windows (S_t contains times $t, t+1$, while S_{t+1} contains $t+1, t+2$, *etc.*); 2) the alignment net's inductive smoothness bias, whereby similar images yield similar predicted transformations; and 3) the temporal alignment terms in the losses.

4 Multiview Moving Camera Dataset (MMCD)

To our knowledge, no available dataset supports multiple video stitching with $N > 2$ freely moving cameras. Thus, we created MMCD, a new multiview dataset. MMCD contains 53 scenes, each with 3–6 approximately synchronized handheld videos captured on diverse mobile devices under varied environmental and lighting conditions. Camera models, resolutions, and motions vary substantially, providing a broad evaluation of MViS. Overall, MMCD contains 221 videos and 47,040 frames. It is divided into 4 categories based on N, the number of cameras: 22 scenes with $N = 3$, 10 with $N = 4$, 11 with $N = 5$, and 10 with $N = 6$. Same-scene videos have the same length, with recordings between 5 and 10 seconds. For video-length statistics across different values of N and overlap statistics, see § E.4. To facilitate quantitative evaluation, we manually annotated keypoints in the overlap regions of the first frames and propagated them through the sequences using CoTracker3 [38]. The annotations, used solely for evaluation, are available with the dataset.

5 Experiments and Results

In all experiments, we optimize MViS using ADAM with a ReduceLROnPlateau scheduler and learning rates of 5×10^{-4} (Stage 1) and 5×10^{-5} (Stage 2). Batch size is 6 for $N = 2$ and 4 for $N > 2$, and each stage is optimized for 600 epochs on an RTX 6000 GPU. We use RoMa [26] as the image matcher; MViS is not tied to this choice, and in § C.3 we report results using LoFTR [27].

Table 1: $N = 2$. Comparison across the 4 categories in the StabStitch-D dataset [1]. The PSNR (↑) and SSIM (↑) values were averaged over all of the frames in the category.

Method	Regular		Low-Light		Low-Texture		Fast-Moving	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
LCP [39]	24.22	.812	-	-	-	-	23.88	.813
UDIS++ [21]	23.19	.785	31.09	.936	29.98	.906	21.56	.756
StabStitch [1]	24.64	.832	34.51	.958	33.63	.927	23.36	.787
StabStitch++ [2]	**25.51**	.837	35.10	.958	**34.23**	.928	25.64	.830
MViS (Ours)	25.47	**.846**	**36.51**	**.960**	30.80	**.932**	**25.76**	**.850**

In Stage 2, we use normalized RGB rather than deep features; DINO [25] features did not improve results in our experiments (likely due to their lower spatial resolution) and increased memory cost.

We also provide videos in the Supplemental Material and at our project page, https://bgu-cs-vil.github.io/MViS-project/, to demonstrate MViS's temporal coherence and handling of time-varying view-intersection graphs.

For $\mathbf{N = 2}$, we evaluate on the StabStitch-D benchmark [1] and compare with LCP [39], UDIS++ [21], StabStitch [1], and StabStitch++ [2]. We report Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM). Table 1 shows that MViS achieves the best SSIM in all categories and the best PSNR in 2/4; all non-MViS numbers are taken from [1, 2]. The '-' symbol indicates that LCP failed in that category.

For $\mathbf{N > 2}$, no existing video stitching method handles freely moving cameras: StabStitch++ supports only $N = 3$ and requires fixed, known camera ordering. We therefore construct a new benchmark for this setting, based on the annotated MMCD.

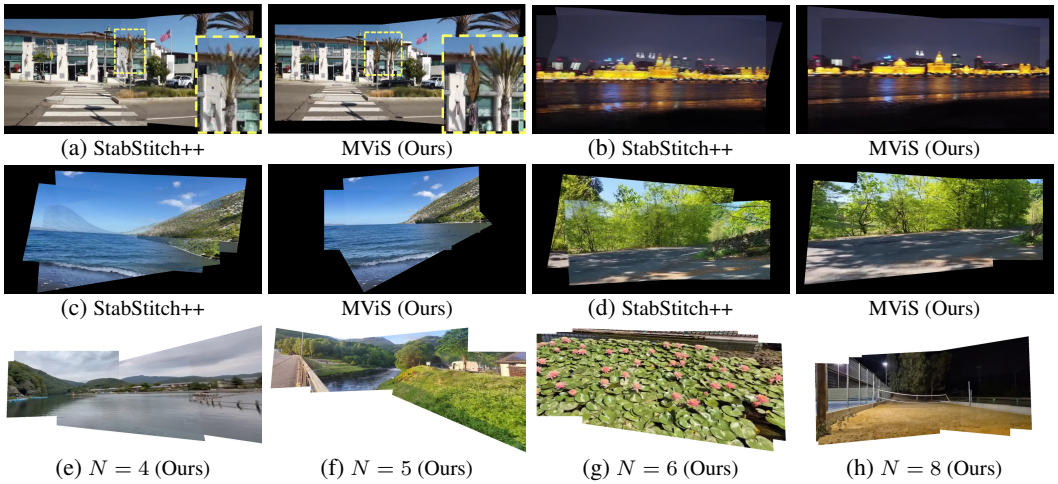
5.1 The Multi-Camera Benchmark

Metrics. For $N > 2$, where PSNR and SSIM are not directly applicable, we report the stability score [14], computed on annotated keypoint trajectories, and our new *Geometric Variance* (GV) metric, which quantifies cross-view consistency of aligned keypoints (details in § E.1). We measure temporal smoothness using the Mean Square Acceleration of video-corner trajectories (CMSA).

As **baselines**, we evaluate MIS methods and video-adapted extensions we built. Since some methods, but not MViS, collapse on certain scenes, we also report the number of successful scenes (Succ.) in each category. For each method, metrics are averaged only over successful scenes, slightly favoring

Table 2: The Multi-Camera Benchmark. Results on the MMCD Dataset. N is the # of videos per scene. The values shown, Stab. (↑), GV (↓) and CMSA (↓), are averaged over the scenes (on which the respective method succeeded without collapsing) with the same N. StabStitch++'s high GV (27.8) on $N = 3$ reflects severe misalignment in scenes where the assumed camera ordering is violated. StabStitch++'s code does not support $N > 3$.

	Method	$N = 3$				$N = 4$				$N = 5$				$N = 6$			
		GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.
MIS	AutoStitch [4]	3.3	.847	635	14/22	**2.4**	.901	217	6/10	3.1	.834	>1k	9/11	3.3	.924	111	5/10
	APAP [9]	2.9	.850	>1k	19/22	5.	.837	>1k	9/10	2.4	.846	>1k	**11/11**	4.0	.875	>1k	8/10
	GSP [5]	2.8	.648	>1k	19/22	2.5	.649	>1k	9/10	**2.1**	.625	>1k	**11/11**	3.0	.672	>1k	9/10
	SPW [6]	4.8	.807	>1k	**22/22**	10.4	.716	>1k	**10/10**	2.5	.835	>1k	10/11	14.8	.745	>1k	**10/10**
	GES-GSP [7]	2.9	.618	>1k	19/22	2.5	.668	>1k	9/10	2.4	.614	>1k	10/11	3.2	.670	>1k	9/10
	OBJ-GSP [8]	2.9	.621	>1k	19/22	2.5	.634	>1k	9/10	2.2	.683	>1k	8/11	3.1	.681	>1k	6/10
MIS+TR	GSP+TR	3.1	.863	62.9	19/22	2.8	.898	52.2	9/10	3.2	.817	93.6	**11/11**	3.5	.879	38.1	9/10
	GES-GSP+TR	3.3	.849	69.4	19/22	3.1	.884	57.2	9/10	3.8	.800	100	**11/11**	3.9	.866	39.5	9/10
	OBJ-GSP+TR	3.2	.853	75.3	19/22	3.0	.879	71.6	9/10	3.5	.808	106	**11/11**	3.7	.872	42.0	9/10
MIS+PS	GSP+PS	**2.7**	.816	216	19/22	**2.4**	.841	251	9/10	**2.1**	.783	577	**11/11**	**2.8**	.867	174	9/10
	GES-GSP+PS	2.8	.802	206	19/22	2.5	.853	154	9/10	2.3	.796	644	10/11	3.0	.887	90.7	9/10
	OBJ-GSP+PS	2.8	.814	177	19/22	**2.4**	.836	196	9/10	**2.1**	.822	73.6	8/11	2.9	.884	101	6/10
VS	StabStitch++ [2]	27.8	.799	20.0	**22/22**	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	MViS (Ours)	3.1	**.917**	**2.5**	**22/22**	2.8	**.925**	**14.2**	**10/10**	2.3	**.874**	**11.**	**11/11**	3.0	**.930**	**2.4**	**10/10**



(a) StabStitch++ MViS (Ours) (b) StabStitch++ MViS (Ours)

(c) StabStitch++ MViS (Ours) (d) StabStitch++ MViS (Ours)

(e) $N = 4$ (Ours) (f) $N = 5$ (Ours) (g) $N = 6$ (Ours) (h) $N = 8$ (Ours)

Figure 6: **Visual comparisons.** *Top row:* $N = 2$ cases from the StabStitch-D dataset [1]. StabStitch++'s severe misalignment makes the traffic sign vanish (a) and causes ghosting of the buildings (b). *Middle:* $N = 3$ cases from the MMCD dataset. *Bottom:* MViS results on MMCD cases with $N = 4, 5, 6, 8$. The $N = 8$ is an extended scene from the $N = 6$ category.

baselines because their failures are typically on hard scenes; full results and the detailed evaluation protocol appear in the Appendix. We now analyze the results summarized in Table 2.

1. Comparison with MIS: Per-frame MIS methods [4, 9, 5, 6, 7, 8] may produce plausible single-frame panoramas, but their videos exhibit severe temporal inconsistency: their stability is substantially lower than MViS and their CMSA (↓) is orders of magnitude higher. This shows that strong per-frame alignment does not imply temporal coherence.

2. Comparison with temporally extended MIS variants: We augment key MIS methods (GSP, GES-GSP, OBJ-GSP) with two stabilization strategies: (1) temporal regularization penalizing changes in estimated transformations between consecutive frames (TR), and (2) post-hoc smoothing of per-frame transformations (PS). Both improve stability over per-frame baselines but still lose to MViS. This is expected: in multiple moving-camera videos, the optimal stabilization strength is not uniform, since different cameras, and hence different panoramic regions, can have different and possibly time-dependent motions.

3. Comparison with Video Stitching (VS): Among existing video stitching methods, StabStitch++ is the only one supporting more than two inputs, but it is limited to $N = 3$ and requires a fixed spatial ordering of the cameras, as noted in its GitHub repository: *"video1 should overlap with video2, and video2 should overlap with video3"*. This assumes that the cameras' relative ordering remains fixed, whereas in freely moving multi-camera capture the view-intersection graph can change over time. Thus, StabStitch++ is not fully applicable to three moving cameras; in practice, when cameras swap order, its output degrades. MViS outperforms StabStitch++ in both alignment and stability metrics. For $N > 3$, no existing video stitching method is applicable, so we report results only for MViS.

5.2 Qualitative Results, Runtime and Memory Usage

Figure 6 compares MViS with StabStitch++ ($N \in \{2, 3\}$) and shows that MViS maintains visual consistency even under rapid motion. Figure 7 shows two scenes in which the view-intersection graph changes over time: as the cameras move, overlaps between views appear and disappear, and MViS adapts the panorama accordingly while keeping every camera consistently placed throughout the video. Additional visualizations, including $N > 3$ cases, appear in the Supplemental Material and on our project webpage.

Runtime. We report time per frame, computed as total runtime divided by N times the number of frames: Stage 1 takes 0.5–0.6 s, Stage 2 takes 0.7–0.8 s, and preprocessing takes 0.4–0.5 s. These times are stable across different N values.

Memory. Memory usage is independent of the original image resolution and scales linearly with N.

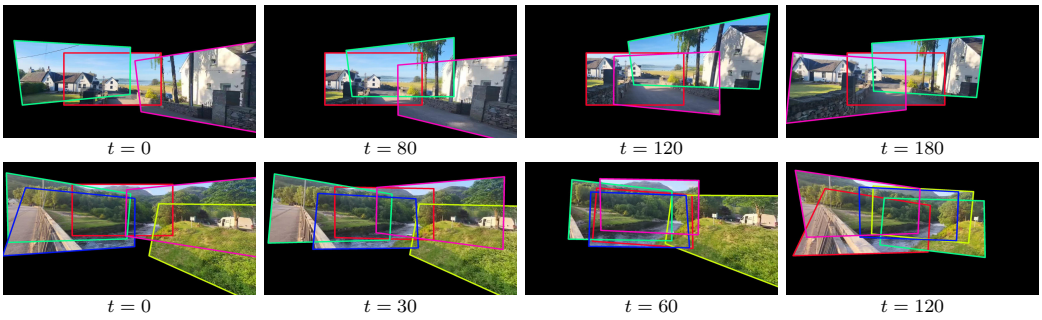


Figure 7: Dynamically changing view-intersection graph, MViS (Ours). Camera outlines are color-coded to highlight overlaps. Each camera is assigned a consistent color throughout the video.

5.3 Ablation Study and Direct Optimization without a Net

Table 3 summarizes an **Ablation Study** on StabStitch-D ($N = 2$) and MMCD ($N > 2$). Removing Stage 1 causes frequent failures, confirming its importance. Removing Stage 2 degrades alignment, especially when inter-view overlap is small. Optimizing the Stage 1 and Stage 2 losses simultaneously usually harms performance, highlighting the need for our two-stage approach. Using unnormalized RGB also hurts performance. As expected, the temporal loss terms mainly affect video stabilization.

To validate the role of the network, we also directly optimize the proposed losses over the homographies, without a neural network. We test three variants: geometric loss only, appearance loss only, and geometric-then-appearance sequential optimization. The resulting videos are spatially aligned but exhibit severe temporal jitter. Adding regularization over homographies could reduce this jitter, but would require per-scene hyperparameter tuning and, more problematically, supervision for selecting appropriate values. In contrast, MViS obtains temporal smoothness implicitly, partly through the inductive smoothness bias of the network (see § 3.7): consecutive frames tend to be similar, and so are the predictions. Since Stability and GV are computed only in overlap regions, where temporal instability is less visible, their numerical differences remain small despite the clear qualitative difference; CMSA captures it well. See the Appendix for visual and quantitative results.

Table 3: Ablation Study. See text for details.

Ablation	$N = 2$		$N = 3$			$N = 4$			$N = 5$			$N = 6$		
	PSNR	SSIM	GV	Stab.	CMSA	GV	Stab.	CMSA	GV	Stab.	CMSA	GV	Stab.	CMSA
No stage 1	25.22	.794	879	.886	>1k	>1k	.888	>1k	>1k	.872	>1k	>1k	.909	>1k
No stage 2	28.93	.897	3.5	.915	>1k	12.5	**.928**	275	2.6	.873	264	3.3	**.930**	10.8
No temporal loss	30.13	.903	**3.1**	.911	44.9	**2.8**	.917	>1k	2.4	.870	48.4	**3.0**	.926	31.3
Both stages simultaneously	25.40	.779	12.9	.905	42.0	11.7	.920	131	4.6	.871	12.7	11.0	**.930**	**0.8**
Unnormalized RGB	29.18	.891	**3.1**	.914	5.8	13.9	.907	31.9	2.6	**.874**	**7.0**	3.1	**.930**	2.5
MViS	**30.24**	**.904**	**3.1**	**.917**	**2.5**	**2.8**	.925	**14.2**	**2.3**	**.874**	11.0	**3.0**	**.930**	2.4

6 Conclusion

We presented MViS, a method for stitching videos from multiple freely moving cameras. MViS combines a regularization-free joint-alignment formulation with a two-stage optimization over geometric and appearance consistency, enabling accurate stitching across arbitrary numbers of views and flexible view-intersection graphs. Experiments show clear advantages over existing methods, particularly for $N > 2$, where prior video-stitching methods are either inapplicable or require restrictive fixed-ordering assumptions.

Limitations. 1) Runtime: optimization time would not suffice for real-time applications. 2) Although the relative-composition framework can support richer invertible transformations, such as nonrigid diffeomorphisms, our current implementation uses homographies, a standard and cost-effective choice that already performs well in our experiments. 3) When the image matcher fails, e.g., due to a dominant textureless region, the ORTs may be unreliable and performance can temporarily degrade, although MViS typically recovers once reliable matches reappear later in the video.

We are not aware of any negative societal impacts associated with this work.

Acknowledgments and Disclosure of Funding

Shira Ifergane was partially supported by the VATAT National Excellence Scholarship for MSc Students in AI and Data Science. Roy Amoyal was partially supported by the Kreitman School of Advanced Graduate Studies.

The authors declare no competing interests.

References

[1] Lang Nie, Chunyu Lin, Kang Liao, Yun Zhang, Shuaicheng Liu, Rui Ai, and Yao Zhao. Eliminating warping shakes for unsupervised online video stitching. In *ECCV*. Springer, 2024.

[2] Lang Nie, Chunyu Lin, Kang Liao, Yun Zhang, Shuaicheng Liu, and Yao Zhao. StabStitch++: Unsupervised online video stitching with spatiotemporal bidirectional warps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[3] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2015.

[4] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 2007.

[5] Yu-Sheng Chen and Yung-Yu Chuang. Natural image stitching with the global similarity prior. In *ECCV*, 2016.

[6] Tianli Liao and Nan Li. Single-perspective warps in natural image stitching. *IEEE Transactions on Image Processing*, 2020.

[7] P. Du, J. Ning, J. Cui, S. Huang, X. Wang, and J. Wang. Geometric structure preserving warp for natural image stitching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2022.

[8] Wenxiao Cai and Wankou Yang. Object-level geometric structure preserving for natural image stitching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[9] Julio Zaragoza, Tat-Jun Chin, Michael S Brown, and David Suter. As-projective-as-possible image stitching with moving DLT. In *CVPR*. IEEE, 2013.

[10] Jing Li, Zhengming Wang, Shiming Lai, Yongping Zhai, and Maojun Zhang. Parallax-tolerant image stitching based on robust elastic warping. *IEEE Transactions on Multimedia*, 2018.

[11] Ameer Hamza, Rehan Hafiz, Muhammad M. Khan, Yongju Cho, and Jihun Cha. Stabilization of panoramic videos from mobile multi-camera platforms. *Image and Vision Computing*, 2015.

[12] Kaimo Lin, Shuaicheng Liu, Loong-Fah Cheong, and Bing Zeng. Seamless video stitching from hand-held camera inputs. *Computer Graphics Forum*, 2016.

[13] Heng Guo, Shuaicheng Liu, Tong He, Shuyuan Zhu, Bing Zeng, and Moncef Gabbouj. Joint video stitching and stabilization from moving cameras. *IEEE Transactions on Image Processing*, 2016.

[14] Yongwei Nie, Tan Su, Zhensong Zhang, Hanqiu Sun, and Guiqing Li. Dynamic video stitching via shakiness removing. *IEEE Transactions on Image Processing*, 2018.

[15] Tan Su, Yongwei Nie, Zhensong Zhang, Hanqiu Sun, and Guiqing Li. Video stitching for handheld inputs via combined video stabilization. In *SIGGRAPH ASIA Technical Briefs*. Association for Computing Machinery, 2016.

[16] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM transactions on graphics (TOG)*, 2013.

[17] Federico Perazzi, Alexander Sorkine-Hornung, Henning Zimmer, Peter Kaufmann, Oliver Wang, Scott Watson, and Markus Gross. Panoramic video from unstructured camera arrays. *Computer Graphics Forum*, 2015.

[18] Wei Jiang and Jinwei Gu. Video stitching with spatial-temporal content-preserving warping. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2015.

[19] Wei-Sheng Lai, Orazio Gallo, Jinwei Gu, Deqing Sun, Ming-Hsuan Yang, and Jan Kautz. Video stitching for linear camera arrays. In *BMVC*, 2019.

[20] I-Chan Lo, Kuang-Tsu Shih, and Homer H Chen. Efficient and accurate stitching for 360° dual-fisheye images and videos. *IEEE Transactions on Image Processing*, 2022.

[21] Lang Nie, Chunyu Lin, Kang Liao, Shuaicheng Liu, and Yao Zhao. Parallax-tolerant unsupervised deep image stitching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[22] Dolev Ofri-Amar, Michal Geyer, Yoni Kasten, and Tali Dekel. Neural congealing: Aligning images to a joint semantic atlas. In *CVPR*, 2023.

[23] Kamal Gupta, Varun Jampani, Carlos Esteves, Abhinav Shrivastava, Ameesh Makadia, Noah Snavely, and Abhishek Kar. ASIC: Aligning sparse in-the-wild image collections. In *ICCV*, 2023.

[24] Nir Barel, Ron Shapira Weber, Nir Mualem, Shahaf E Finder, and Oren Freifeld. SpaceJAM: a lightweight and regularization-free method for fast joint alignment of images. In *ECCV*. Springer, 2024.

[25] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[26] Johan Edstedt, Qiyu Sun, Georg Bökman, Mårten Wadenbäck, and Michael Felsberg. RoMa: Robust dense feature matching. In *CVPR*, 2024.

[27] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. LoFTR: Detector-free local feature matching with transformers. In *CVPR*, 2021.

[28] Christopher Mei, Selim Benhimane, Ezio Malis, and Patrick Rives. Homography-based tracking for central catadioptric cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2006.

[29] Federica Arrigoni and Andrea Fusiello. Synchronization problems in computer vision with closed-form solutions. *International Journal of Computer Vision*, 2020.

[30] Omri Hirsch, Ron Shapira Weber, Shira Ifergane, and Oren Freifeld. FastJAM: a fast joint alignment model for images. In *NeurIPS*, 2025.

[31] Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *CVPR*, 2018.

[32] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

[33] Ron Shapira Weber and Oren Freifeld. Regularization-free diffeomorphic temporal alignment nets. In *ICML*. PMLR, 2023.

[34] Avihai Naaman, Ron Shapira Weber, and Oren Freifeld. Synchronization of multiple videos. In *ICCV*, 2025.

[35] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *The International Journal of Robotics Research*, 2019.

[36] Mark Cox, Sridha Sridharan, Simon Lucey, and Jeffrey Cohn. Least-squares congealing for large numbers of images. In *ICCV*. IEEE, 2009.

[37] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI*, 1981.

[38] Nikita Karaev, Yuri Makarov, Jianyuan Wang, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. CoTracker3: Simpler and better point tracking by pseudo-labelling real videos. In *ICCV*, 2025.

[39] Qi Jia, ZhengJun Li, Xin Fan, Haotian Zhao, Shiyu Teng, Xinchen Ye, and Longin Jan Latecki. Leveraging line-point consistence to preserve structures for wide parallax image stitching. In *CVPR*, 2021.

Video Stitching from Multiple Moving Cameras

Appendix

In this appendix, we provide additional qualitative and quantitative results, implementation details, further information on the proposed metric, and dataset length statistics.

In addition to this document, we include a set of videos under `videos/`. The structure and content of the videos are described in § A.

Content:

This document contains the following:

Appendix A. Additional Qualitative Results:

1) An Overview of the Supplementary Videos
2) Comparisons on the $N = 3$ Case (Fig. 8)
3) MViS panoramas with dynamically changing overlap graphs (Fig. 7).
4) Comparison to Multiple Image Stitching (MIS) in $N > 2$ (Fig. 9)

Appendix B. Parameterizing Homographies via the Special Linear Group

Appendix C. Additional Experiments:

§ C.1. Optimization without a Neural Network

§ C.2. Regularization Experiments

§ C.3. Ablation on Image Matcher Choice

§ C.4. Effect of Camera Motion

§ C.5. Robustness to ORT Corruption

Appendix D. The Alignment Network Architecture

Appendix E. Further Details on the Multi-Camera Benchmark and MMCD Dataset:

§ E.1. The Geometric Variance Metric (GV)

§ E.2. The Multi-Camera Benchmark Evaluation Protocol

§ E.3. Implementation details of the Extended MIS Variants

§ E.4. Additional MMCD Statistics (Fig. 11, Table 11)

§ E.5. Full Multi-Camera Benchmark Results (Table 12-17)

In addition, key notational conventions used across the main paper are summarized in Table 18.

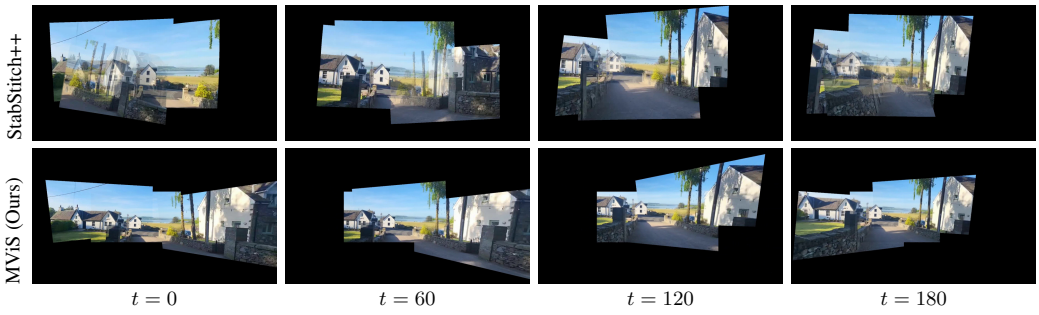
A Additional Qualitative Results

An Overview of the Supplementary Videos

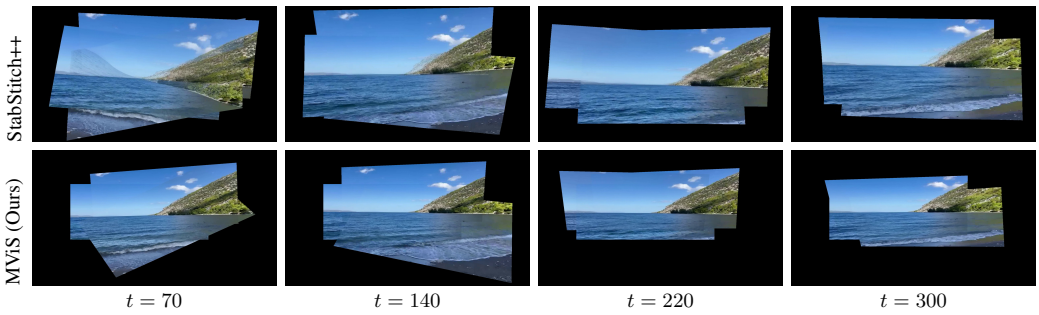
We include a set of videos in the Supplementary Material:

- ▷ `2_stabstitch++_vs_mvis.mp4`,
  `3_stabstitch++_vs_mvis.mp4`,
  `4_stabstitch++_vs_mvis.mp4`
  Visual comparisons between StabStitch++ and MViS (Ours) on $N = 3$, MMCD. The specific challenges of these scenes are illustrated in Fig. 8 of this appendix.
- ▷ `5_n5_mis_methods_vs_mvis.mp4`
  `6_n4_mis_methods_vs_mvis.mp4`
  Visual comparison between Multiple Image Stitching (MIS) methods and MViS on scenes from the MMCD dataset with $N = 4, 5$ cameras. While each MIS method produces visually plausible panoramas per frame, their resulting panoramic video exhibits strong temporal instability.
- ▷ `7_mis_variants_vs_mvis.mp4`
  Visual comparison between OBJ-GSP, OBJ-GSP+TR, OBJ-GSP+PS and MViS. Although OBJ-GSP produces visually plausible panoramas per frame, the resulting video is temporally unstable. Both extensions improve temporal stability but still underperform MViS.
- ▷ `8_n5_mvis.mp4`
  MViS video panorama of an $N = 5$ scene (that appears in Fig. 6 in the paper and Fig. 7) from the MMCD dataset, demonstrating robustness to dynamically changing overlap topology.
- ▷ `9_n2_stabstitch++_vs_mvis.mp4`,
  `10_n2_stabstitch++_vs_mvis.mp4`
  Visual comparisons for $N = 2$ on StabStitch-D (corresponding to scenes from Fig. 6 in the paper).
- ▷ `11_with_vs_without_net.mp4`
  Visual comparison between MViS and a direct optimization of the homographies without a neural network. While it can achieve good results in terms of per-frame panoramas, it yields, usually, much less temporally stable results, as is evident in the video.
- ▷ `12_mmcd_sample.mp4`,
  Samples of annotated scenes from the MMCD dataset.
- ▷ `13_mvis_failure_case.mp4`
  MViS failure case discussed in the paper, where the matcher fails in a texture-less scene (white wall).

The relative left-right ordering of the cameras is not fixed; the cameras swap positions over time.



Challenging motion from three independent moving cameras and extensive moving water in the scene.



Very low inter-camera overlap and dynamically changing spatial ordering.

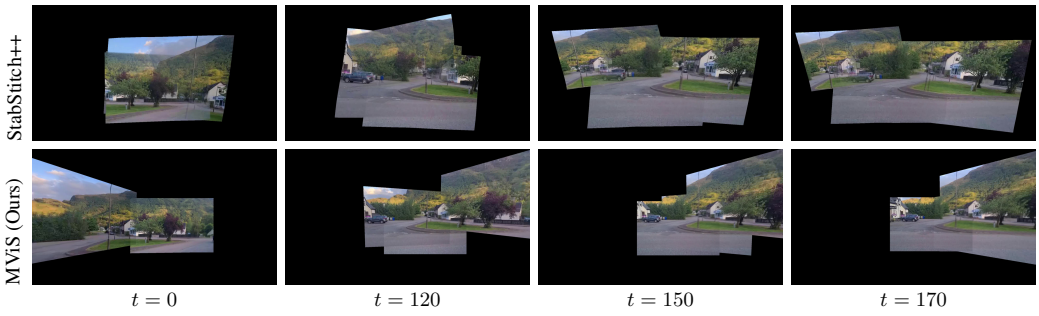


Figure 8: Visual comparison between StabStitch++ [2] and MViS scenes with $N = 3$ cameras (from the MMCD dataset).

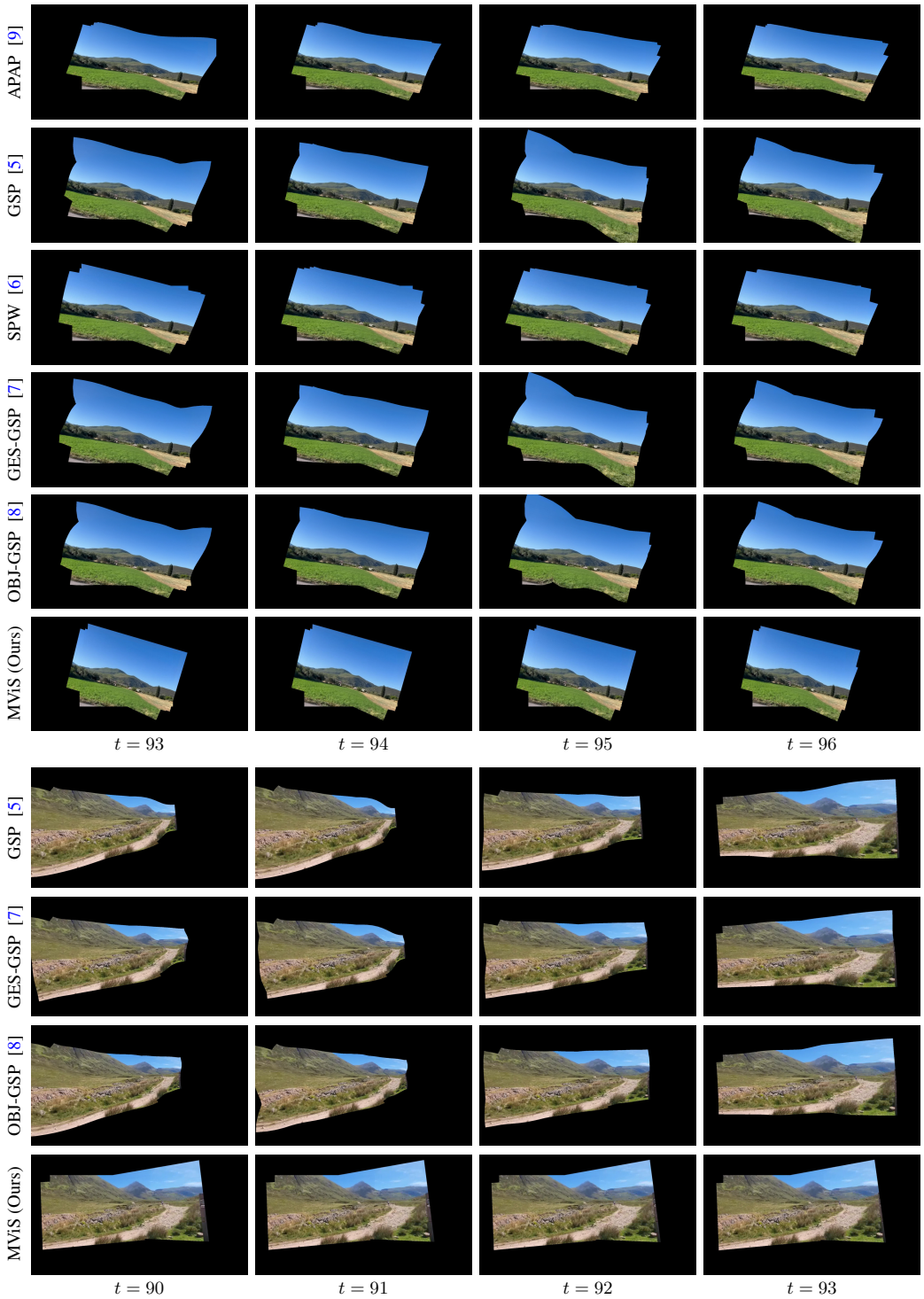


Figure 9: Visual comparison between MIS methods and MViS on scenes with $N > 2$ cameras from the MMCD dataset. We show a sequence of consecutive timestamps to illustrate that, although MIS methods often produce high-quality panoramas at individual frames, they lack temporal consistency.

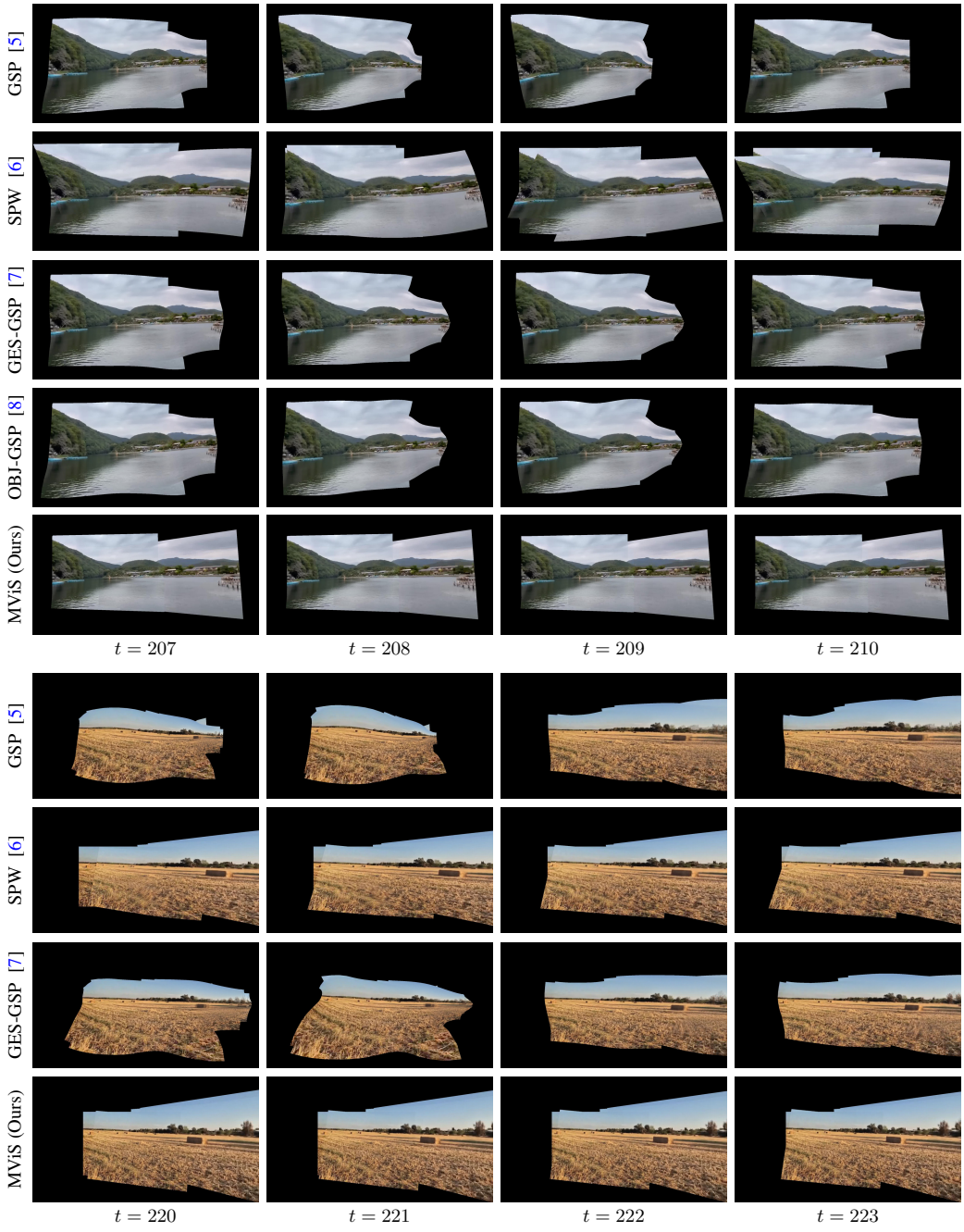


Figure 10: Visual comparison between MIS methods and MViS on scenes with $N > 3$ cameras from the MMCD dataset, continued.

B Parameterizing Homographies via The Special Linear Group

Without special care, a deep net might predict non-invertible matrices. Besides the fact that this could hinder the optimization process [31], our approach requires invertible matrices. Thus, we use the well-known parameterization of homographies using the Special Linear group $\mathrm{SL}(3)$ (see, *e.g.*, [28, 29, 24]). Recall that a homography has 8 degrees of freedom and every homography is identified with an equivalence class of invertible matrices, where all of the class members are proportional to each other and the class representative can be taken to be that whose determinant is one. Let

$$\begin{aligned}\mathfrak{sl}(3) &= \left\{ \boldsymbol{\Theta} : \boldsymbol{\Theta} = \begin{bmatrix} \theta_1 & \theta_2 & \theta_3 \\ \theta_4 & \theta_5 & \theta_6 \\ \theta_7 & \theta_8 & -(\theta_1+\theta_5) \end{bmatrix} \right\}, \\ \mathrm{SL}(3) &= \left\{ \boldsymbol{H} : \boldsymbol{H} = \begin{bmatrix} H_1 & H_2 & H_3 \\ H_4 & H_5 & H_6 \\ H_7 & H_8 & H_9 \end{bmatrix} \,\&\, \det \boldsymbol{H} = 1 \right\}. \end{aligned} \tag{9}$$

$\mathrm{SL}(3)$ is the matrix Lie group of real-valued 3-by-3 matrices with determinant 1, while $\mathfrak{sl}(3)$ is the corresponding Lie algebra of zero-trace real-valued 3-by-3 matrices. While $\mathfrak{sl}(3)$ is a linear space and $\mathrm{SL}(3)$ is a nonlinear manifold, both SL and $\mathfrak{sl}$ have the same dimension, 8. Moreover, $\mathrm{SL}(3)$ can be identified with the group of homographies in $\mathbb{R}^2$ and the matrix exponential maps $\mathrm{sl}(3) \to \mathrm{SL}(3)$ via $\boldsymbol{\Theta} \mapsto \exp(\boldsymbol{\Theta})$. Thus, we set our alignment network to predict zero-trace matrices by predicting 8-length vectors of the form $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_8)$, where the obvious bijection between the vector $\boldsymbol{\theta}$ and the zero-trace 3-by-3 matrix $\boldsymbol{\Theta}$ is as in Eq. 9. Then, via the matrix exponential, we obtain homographies. In a slight abuse of notation, we will write $\exp(\boldsymbol{\theta})$ instead of $\exp(\boldsymbol{\Theta})$. Also useful are the following two facts: 1) a zero-vector $\boldsymbol{\theta} = \mathbf{0}_{8\times 1}$ corresponds to the identity matrix (i.e., $\boldsymbol{H} = \boldsymbol{I}_{3\times 3}$), and 2) the inverse matrix of $\exp(\boldsymbol{\theta})$ is, conveniently, $\exp(-\boldsymbol{\theta})$. Finally, for two different 8-length vectors, $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$, in general $\exp(\boldsymbol{\theta}_i)\exp(\boldsymbol{\theta}_j) \neq \exp(\boldsymbol{\theta}_j)\exp(\boldsymbol{\theta}_i)$ (i.e., the homography group is not Abelian); this detail means that the order in which composition of the transformation is done (*e.g.*, in our loss functions) is important.

Remark: Recall that in general any invertible matrix defines a homography, and that if that matrix is multiplied by a nonzero scalar it still defines the same homography. In other words, in general a homography corresponds to an equivalence class of invertible matrices (that are proportional to each other). However, when using the SL parameterization above, this becomes a nonissue since, by construction, the $\det = 1$ constraint essentially boils down to picking, for each such equivalence class, exactly one representative: the one whose determinant is 1.

C Additional Experiments

C.1 Optimization without a Neural Network

Table 4 reports CMSA values (lower is better) for the **Optimization without a net** experiment described in the main paper. Instead of predicting homography parameters with the Alignment Network, we **directly optimize** the per-frame homography parameters $\{\boldsymbol{\theta}_i^t\}$ under the same losses, considering three variants:

1. $\mathcal{L}_{\text{geo}}$ (w/o Net) - Geometric Loss only
2. $\mathcal{L}_{\text{rgb}}$ (w/o Net) - Appearance Loss only
3. w/o Net - a two stage scheme that first minimizes $\mathcal{L}_{\text{geo}}$ and then uses the resulting solution as initialization to minimize $\mathcal{L}_{\text{rgb}}$.

To ensure proper convergence, for each variant we test 4 learning rate values and report the best result; each loss is optimized for 1000 iterations (so w/o Net runs 2000 iterations in total).

Across all N values, MViS consistently achieves lower CMSA than any optimization-only variant, showing that the Alignment Network contributes to temporal stabilization. As explained in the paper, we attribute this phenomenon to the inductive bias of the network towards smoother solutions: consecutive frames tend to be similar, so they lead to similar predicted homographies. A visual comparison between MViS and a direct optimization without a neural network is provided in `11_with_vs_without_net.mp4`.

Table 4: **Optimization without a net.** Measured CMSA on the frame's corners. In $N = 2$ we used StabStitch-D dataset, and for $N > 2$ the MMCD dataset.

Loss	$N = 2$	$N = 3$	$N = 4$	$N = 5$	$N = 6$
$\mathcal{L}_{geo}$ (w/o Net)	39.9	>1k	>1k	485.3	18.0
$\mathcal{L}_{\text{rgb}}$ (w/o Net)	>1k	>1k	>1k	>1k	>1k
w/o Net	118.7	35.8	86.4	11.6	21.3
MViS	**6.8**	**2.5**	**14.2**	**11.**	**2.4**

C.2 Regularization Experiments

We experimented with adding explicit regularization terms to the loss formulations of both stages, weighted by a global coefficient α that controls the strength of the regularization. We evaluated two terms over the predicted transformations using three different global weights $\alpha \in \{0.5, 1, 5\}$.

The first term, $\mathcal{R}_1 = \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{\Theta}_i^{t+1} - \mathbf{\Theta}_i^{t} \right\|_F^2$, encourages temporal smoothness by penalizing changes in the predicted transformation parameters between consecutive frames. The second, $\mathcal{R}_2 = \frac{1}{2N} \sum_{k \in \{0,1\}} \sum_{i=1}^{N} \left\| \mathbf{\Theta}_i^{t+k} \right\|_F^2$, penalizes the magnitude of the predicted transformations. $\mathbf{\Theta}$ is defined in Eq. 9. We evaluated these variants on two datasets: StabStitch-D ($N = 2$ case) and the MMCD dataset ($N = 3, 4$ cases).

As shown in Table 5, explicit regularization did not provide a consistent overall improvement over MViS across the evaluated datasets, categories, and metrics. This of course does not suggest that regularization is always inherently harmful; however, the point is that MViS simply does not require regularization terms to perform well. Moreover, making such regularization effective requires selecting a suitable value of α, and choosing this value in practice requires supervision, since one must evaluate which setting performs best. Beyond the fact that the optimal α may vary across scenes, and potentially even across cameras and time, this tuning process introduces an additional supervised model-selection step. Avoiding this dependency is therefore a practical advantage of MViS.

Table 5: Regularization experiments on StabStitch-D dataset ($N = 2$) and the MMCD dataset ($N = 3, 4$).

$\mathcal{R}$	α	$N=2$		$N=3$			$N=4$		
		PSNR ↑	SSIM ↑	GV ↓	Stab. ↑	CMSA ↓	GV ↓	Stab. ↑	CMSA ↓
$\mathcal{R}_1$	0.5	**30.24**	**.904**	**3.1**	**.917**	2.2	**2.8**	**.925**	35.1
	1.0	30.17	**.904**	**3.1**	**.917**	2.8	**2.8**	**.925**	14.2
	5.0	29.97	.901	**3.1**	.916	**1.9**	**2.8**	.924	15.6
$\mathcal{R}_2$	0.5	29.02	.890	7.2	.913	21.7	6.7	**.925**	16.6
	1.0	25.12	.819	14.3	.909	15.9	24.3	.894	**10.6**
	5.0	19.10	.613	44.8	.881	36.6	86.2	.838	113.2
MViS	–	**30.24**	**.904**	**3.1**	**.917**	2.5	**2.8**	**.925**	14.2

C.3 Ablation on Image Matcher Choice

To evaluate the robustness of MViS to the choice of image matcher, we run our method with RoMa [26] and LoFTR [27]. All other components, optimization settings, and hyperparameters are kept fixed. As shown in Table 6, MViS performs reliably with both matchers, indicating that our framework does not depend on a specific matcher, but only requires sufficiently reliable pairwise correspondences.

Table 6: Image Matchers Comparison.

Ablation	$N=2$		$N=3$			$N=4$			$N=5$			$N=6$		
	PSNR	SSIM	GV	Stab.	CMSA	GV	Stab.	CMSA	GV	Stab.	CMSA	GV	Stab.	CMSA
MViS (LoFTR)	30.19	**.905**	**3.1**	**.917**	3.5	**2.8**	.924	**12.4**	**2.3**	**.874**	**8.7**	**3.0**	**.930**	4.3
MViS (RoMa)	**30.24**	.904	**3.1**	**.917**	**2.5**	**2.8**	**.925**	14.2	**2.3**	**.874**	11.0	**3.0**	**.930**	**2.4**

C.4 Effect of Camera Motion

To study how camera motion affects MViS, we assign each MMCD scene a scalar camera-motion score.

Camera motion. For each time t, we compute the largest corner displacement across all cameras:

$$m_t = \max_{i=1,\dots,N} \max_{k=1,\dots,4} \left\| \pi\left(\widehat{\boldsymbol{H}}_i^{t\to t+1} \tilde{\mathbf{c}}_k\right) - \mathbf{c}_k \right\|_2, \tag{10}$$

where $\widehat{\boldsymbol{H}}_i^{t\to t+1}$ is an ORT, $\mathbf{c}_k$ is the k-th image corner, $\tilde{\mathbf{c}}_k$ is its homogeneous representation, and π converts it to Euclidean coordinates.

The scene-level camera motion is defined as

$$\text{CameraMotion} = 100 \cdot \underset{t}{\text{median}} \left(\frac{m_t}{\text{image diagonal}} \right). \tag{11}$$

We partitioned the MMCD scenes into 3 groups by camera-motion score and report the median and IQR for each metric in Table 7. Higher camera motion is associated with lower Stab. and higher CMSA, while GV shows no monotonic trend, indicating no clear association with geometric alignment quality in these experiments.

Table 7: Effect of camera motion on MViS

Group	CameraMotion range	# of scenes	GV ↓	Stab. ↑	CMSA ↓
Low	0.17–0.46	17	2.3 [1.7–3.0]	.927 [.913–.958]	0.1 [0.1–0.5]
Mid	0.48–0.77	16	3.1 [2.1–3.4]	.927 [.913–.940]	1.0 [0.3–7.1]
High	0.78–1.50	20	2.4 [1.6–3.2]	.896 [.852–.928]	5.3 [1.1–14.2]

C.5 Robustness to ORT Corruption

We perturb 10%, 20%, or 30% of the temporal and multi-view ORTs using projective perturbation matrices constructed from eight Gaussian parameters ($\sigma = 0.01$ for affine/translation components and 0.0015 for projective components), following Appendix B, Eq. (9).

Because each mask is derived by warping an all-ones mask with its estimated ORT, perturbing an ORT also perturbs the corresponding mask. Independently perturbing the masks would decouple them from the ORTs and would not reflect the pipeline's actual failure mode.

Since scenes with little shared content between views are expected to be more sensitive to ORT errors, we quantify each scene's overlap as follows.

Overlap ratio. For each directed mask $M^t_{i\to j}$, let

$$O^t_{i\to j} = \frac{\text{number of white pixels in } M^t_{i\to j}}{\text{total number of pixels}} \tag{12}$$

denote the fraction of white pixels in the mask.

We then compute the weakest pairwise overlap at each time step, $B_t = \min_{i\neq j} O^t_{i\to j}$, and define the scene-level overlap ratio as $\mathrm{OverlapRatio} = \mathrm{median}_t\ B_t$.

We group the 52 successful MMCD scenes into three categories based on their overlap ratios and report the mean for each category in Table 8. The remaining scene, whose minimum pairwise overlap is only 1.7%, fails under the injected perturbations and is therefore not included in the averages.

Overall, MViS remains robust to ORT corruption. The first-stage loss increases monotonically with the corruption level, confirming that the injected perturbations meaningfully affect the geometric optimization. Specifically, the median first-stage loss across all dataset scenes increases by 70% when the corruption level rises from 10% to 20%, and by 140% from 10% to 30%. In contrast, the median second-stage loss increases by only 0.5% and 1.%, respectively, indicating that most of the effect of the corrupted ORTs is handled during the first-stage optimization, while the subsequent appearance-refinement stage remains stable.

Table 8: Results by overlap-ratio group and noise level.

Group	OverlapRatio range	Noise (%)	GV ↓	Stab. ↑	CMSA ↓	# of scenes
Low	0.27 – 0.52	0	2.9	.914	10.7	18
		10	2.9	.913	7.2	
		20	2.9	.912	11.0	
		30	3.0	.913	10.1	
Mid	0.52 – 0.66	0	2.7	.914	6.6	16
		10	2.7	.912	6.4	
		20	2.6	.912	6.6	
		30	2.7	.912	6.3	
High	0.66 – 0.85	0	3.1	.912	0.3	18
		10	3.1	.911	0.4	
		20	3.0	.910	0.5	
		30	3.1	.910	0.5	

D The Alignment Network Architecture

Following the parameterization described in Appendix B, the alignment network predicts an 8-dimensional Lie-algebra vector for each camera and time step. We initialize all weights in the final fully connected layer (FC2) to zero, so that the initial predicted transformations are the identity.

The number of parameters in the CNN module is independent of the number of cameras N: all BatchNorm and Conv2d layers operate on $2BN$ input frames and contain only a few thousand parameters. The dependence on N appears only in the fully connected module, whose parameter count grows linearly with N ($10{,}240N{+}32$ for FC1 and $264N$ for FC2).

Table 9: Alignment Network Summary.

Layer (type:depth-idx)	Output Shape	#Params
Alignment Net	$[2B, 8N]$	–
In_Norm (BatchNorm2d): 1-1	$[2BN, 3, 256, 256]$	6
CNN: 1-2	$[2BN, 5, 8, 8]$	–
Conv2d: 2-1	$[2BN, 10, 250, 250]$	1,480
BatchNorm2d: 2-2	$[2BN, 10, 250, 250]$	20
AdaptiveMaxPool2d: 2-3	$[2BN, 10, 32, 32]$	–
ReLU: 2-4	$[2BN, 10, 32, 32]$	–
Conv2d: 2-5	$[2BN, 5, 28, 28]$	1,255
BatchNorm2d: 2-6	$[2BN, 5, 28, 28]$	10
AdaptiveMaxPool2d: 2-7	$[2BN, 5, 8, 8]$	–
ReLU: 2-8	$[2BN, 5, 8, 8]$	–
Flatten (reshape): 1-3	$[2B, 320N]$	–
FC1 (Linear): 1-4	$[2B, 32]$	$10,240N + 32$
GELU: 1-5	$[2B, 32]$	–
FC2 (Linear): 1-6	$[2B, 8N]$	$264N$

E Further Details on the Multi-Camera Benchmark and MMCD Dataset

E.1 The Geometric Variance Metric (GV)

To the best of our knowledge there is no existing suitable metric for multi-view video stitching evaluation when $N > 2$. For example, PSNR and SSIM, while sometimes used in the case of $N = 2$, require a reference frame and a target frame, which are less relevant in the more general case.

To be able to measure the alignment quality of our stitched videos over time, we annotated keypoints on the videos. We stress that these annotations were used only for evaluation purposes.

Concretely, we annotated matching keypoints on the first frame of each video and then used CoTracker3 [38] to predict the trajectories of these keypoints throughout the videos. Let $p_m^{t,i}$ denote the m-th keypoint in F_i^t (*i.e.*, the original frame at time t in video i) as predicted by CoTracker3.

In addition, for every predicted point in every time t, the CoTracker3 model returns a visibility flag $b_m^{t,i}$ which indicates whether the m-th query point is visible in frame t of video i or not. Formally, $b_m^{t,i}$ is 1 if the point is visible in video i at time t and 0 otherwise.

Let n_p be the number of query points. We consider keypoint m at time t as valid only if it is visible in all N videos:

$$M_{m,t} \;=\; \prod_{i=1}^{N} b_m^{t,i}\,. \tag{13}$$

Consequently, the number of valid keypoints in frame t is

$$M_t \;=\; \sum_{m=1}^{n_p} M_{m,t} \tag{14}$$

and the number of frames that contribute to the final score is

$$\widehat{v_l} \;=\; \big|\{\, t \mid M_t > 0 \,\}\big|\,. \tag{15}$$

The per-keypoint geometric variance is calculated across all input videos' frames t:

$$\mathrm{var}_{m,t}[c] \;=\; \frac{1}{N}\sum_{i=1}^{N}\left((T^{t,i}\cdot p_m^{t,i})[c] - \frac{1}{N}\sum_{i=1}^{N}(T^{t,i}\cdot p_m^{t,i})[c]\right)^2, c \in \{1,2\} \tag{16}$$

where $T^{t,i}$ is the transformation that warps frame i of time t to the panoramic space and the index c denotes the coordinate axis. The overall geometric variance score is

$$S_{\mathrm{GV}} \;=\; \frac{1000}{\mathrm{diag}}\cdot\sqrt{\frac{1}{\widehat{v_l}}\sum_{t:\,M_t>0}\frac{1}{M_t}\sum_{m=1}^{n_p} M_{m,t}\,(\overline{\mathrm{var}}_{m,t})} \tag{17}$$

where $\mathrm{diag} = \sqrt{H^2+W^2}$ is the length of the diagonal of the original frames shape (where HxW stands for the original resolution) and $\overline{\mathrm{var}}_{m,t} = \mathrm{var}_{m,t}[1] + \mathrm{var}_{m,t}[2]$. The multiplicative factor $\frac{1000}{\mathrm{diag}}$ is used to account for the fact that videos from different scenes, even within the same dataset, may have different resolutions. As 1000 is about the average length of the diagonal of the frames, we divide the score by the length of the diagonal and multiply by 1000 so each video will influence the score in a similar way, regardless of the resolution.

Naturally, the lower the geometric variance is, the better the result.

Note: The geometric variance metric needs no distance threshold and evaluates all views together in a shared space. We believe it is a more informative alternative to PCK (Percentage of Correct Keypoints) for any task that aligns multiple images into a common frame.

E.2 The Multi-Camera Benchmark Evaluation Protocol

For MIS methods, we run each method independently on every frame. When a method yields panoramas in a drifting coordinate system, we post-align the per-frame outputs to a fixed reference to correct this drift.

If a method fails to produce a panorama for any frame in a scene, we mark the entire scene as collapsed. Quantitative metrics are averaged over non-collapsed scenes only, and we additionally report the number of successfully processed scenes. Full per-scene, per-metric, and per-method results are provided in Table 12- 17.

GSP, GES-GSP, and OBJ-GSP require an adjacency graph over overlapping image pairs and the selection of a reference image. We construct this graph independently at each time step t, and, for improved temporal stability, we consistently choose the first camera as the reference image.

In addition, for these methods, we anchor the resulting panoramas to a shared coordinate system by estimating a similarity transformation that maps the optimized vertices of the reference mesh to its initial vertices, and then applying this same transformation to all mesh vertices of all views at time t. This is conceptually similar to the panorama construction used in MViS.

E.3 Implementation details of the Extended MIS Variants

For evaluation in the video setting, we extended GSP, GES-GSP, and OBJ-GSP in two ways, as their original formulations optimize each frame independently.

Temporal Regularization (TR). We augment the original optimization with a temporal smoothness term that regularizes each mesh vertex in the current frame toward its optimized position in the previous frame, thereby reducing frame-to-frame jitter. Concretely, these temporal constraints are appended as additional equations to the sparse linear system, using the optimized vertices from the previous frame as references.

Post-hoc Smoothing (PS). We also consider a post-processing variant in which the original method is first applied independently to each frame, producing a per-frame set of optimized mesh vertex positions. We then treat each vertex coordinate as a 1D signal over time and apply a Gaussian filter along the temporal axis. Finally, all frames are re-rendered using the smoothed vertex trajectories, yielding a temporally smoother panoramic video.

We experiment with $\sigma \in \{0.5, 1.0, 1.5\}$, where σ is the standard deviation of the Gaussian kernel in PS and $w \in \{0.5, 1.0, 1.5\}$ is the weight of the temporal regularization term in TR. The results are summarized in Table 10. The main paper reports results for the setting $w = 1.0$ and $\sigma = 1.0$. Across all evaluated hyperparameter settings of the MIS extensions, MViS attains the best Stab. and CMSA scores. A visual comparison between OBJ-GSP, OBJ-GSP+TR, OBJ-GSP+PS, and MViS is provided in `7_mis_variants_vs_mvis.mp4`.

Table 10: Additional hyperparameter experiments for the MIS extensions on the MMCD dataset (the Multi-Camera Benchmark). N denotes the # of videos per scene. The values shown, GV (↓), Stab. (↑) and CMSA (↓), are averaged over the scenes (on which the respective method succeeded without collapsing) with the same N.

	Method	$N=3$				$N=4$				$N=5$				$N=6$			
		GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.	GV	Stab.	CMSA	Succ.
$w=0.5$	GSP+TR	2.9	.821	381	19/22	2.5	.872	165	9/10	2.4	.785	297	**11/11**	3.0	.839	204	9/10
	GES-GSP+TR	3.0	.797	393	19/22	2.6	.857	170	9/10	2.9	.768	318	**11/11**	3.3	.824	204	9/10
	OBJ-GSP+TR	3.0	.803	415	19/22	2.6	.842	244	9/10	2.7	.773	339	**11/11**	3.1	.833	213	9/10
$w=1.0$	GSP+TR	3.1	.863	62.9	19/22	2.8	.898	52.2	9/10	3.2	.817	93.6	**11/11**	3.5	.879	38.1	9/10
	GES-GSP+TR	3.3	.849	69.4	19/22	3.1	.884	57.2	9/10	3.8	.800	100	**11/11**	3.9	.866	39.5	9/10
	OBJ-GSP+TR	3.2	.853	75.3	19/22	3.0	.879	71.6	9/10	3.5	.808	106	**11/11**	3.7	.872	42.0	9/10
$w=1.5$	GSP+TR	3.8	.882	21.0	19/22	3.8	.909	22.4	9/10	5.1	.826	39.4	**11/11**	4.9	.901	13.6	9/10
	GES-GSP+TR	4.0	.872	23.7	19/22	4.1	.897	24.9	9/10	5.6	.810	43.0	**11/11**	5.5	.888	14.1	9/10
	OBJ-GSP+TR	4.0	.876	26.7	19/22	4.0	.893	30.6	9/10	5.4	.817	46.7	**11/11**	5.3	.893	16.1	9/10
$\sigma=0.5$	GSP+PS	2.8	.709	>1k	19/22	**2.4**	.722	>1k	9/10	**2.1**	.686	>1k	**11/11**	2.9	.746	>1k	9/10
	GES-GSP+PS	2.8	.685	>1k	19/22	2.5	.741	>1k	9/10	2.4	.683	>1k	10/11	3.1	.754	>1k	9/10
	OBJ-GSP+PS	2.8	.691	>1k	19/22	**2.4**	.709	>1k	9/10	**2.1**	.741	>1k	8/11	3.0	.761	>1k	6/10
$\sigma=1.0$	GSP+PS	**2.7**	.816	216	19/22	**2.4**	.841	251	9/10	**2.1**	.783	577	**11/11**	**2.8**	.867	174	9/10
	GES-GSP+PS	2.8	.802	206	19/22	2.5	.853	154	9/10	2.3	.796	644	10/11	3.0	.887	90.7	9/10
	OBJ-GSP+PS	2.8	.814	177	19/22	**2.4**	.836	196	9/10	**2.1**	.822	73.6	8/11	2.9	.884	101	6/10
$\sigma=1.5$	GSP+PS	**2.7**	.851	43.3	19/22	**2.4**	.880	51.7	9/10	**2.1**	.811	89.3	**11/11**	**2.8**	.898	35.7	9/10
	GES-GSP+PS	2.8	.843	40.6	19/22	2.5	.887	30.0	9/10	2.4	.827	96.2	10/11	3.0	.915	18.3	9/10
	OBJ-GSP+PS	2.8	.855	34.5	19/22	**2.4**	.879	37.0	9/10	2.2	.844	14.8	8/11	2.9	.907	18.3	6/10
	MViS (Ours)	3.1	**.917**	**2.5**	**22/22**	2.8	**.925**	**14.2**	**10/10**	2.3	**.874**	**11.0**	**11/11**	3.0	**.930**	**2.4**	**10/10**

E.4 Additional MMCD Statistics

Figure 11 summarizes the video lengths statistics in MMCD. Table 11 shows the distribution of the 53 MMCD scenes across Overlap ratio intervals. Overlap ratio metric is defined in § C.5.

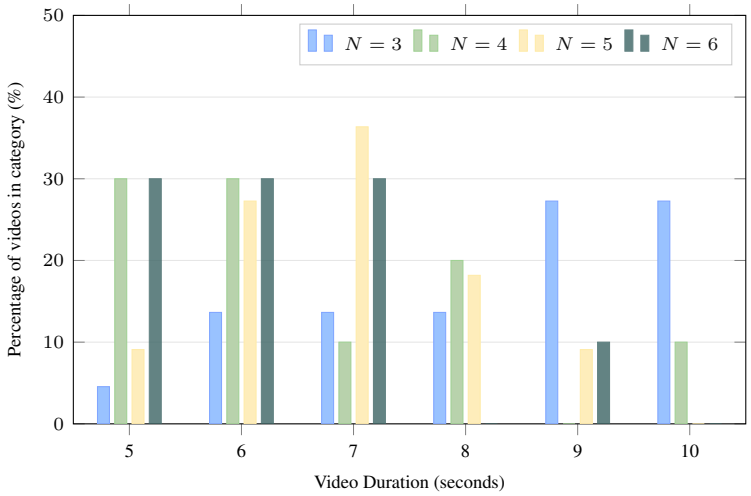


Figure 11: MMCD video duration statistics.

Table 11: Distribution of MMCD scenes across Overlap ratio intervals.

Overlap ratio	0.0–0.1	0.1–0.2	0.2–0.3	0.3–0.4	0.4–0.5	0.5–0.6	0.6–0.7	0.7–0.8	0.8–0.9	0.9–1.0	Total
# of scenes	0	0	3	4	11	12	8	13	2	0	53
Percentage	0.0%	0.0%	5.7%	7.5%	20.8%	22.6%	15.1%	24.5%	3.8%	0.0%	100%

E.5 Full Multi-Camera Benchmark Results

We provide here the complete per-scene benchmark results. In the main paper, the reported averages were computed only over the scenes for which each method successfully generated a panorama. In the tables below, ✗ denotes the method collapsed on the scene, while N/A indicates the method does not support that number of cameras.

Table 12: Per-scene GV(↓), the Multi-Camera Benchmark.

	Data	AutoStitch	APAP	GSP	SPW	GES-GSP	OBJ-GSP	MViS(Ours)
$N = 3$	scene_1	✗	1.7	1.7	1.8	1.7	1.7	2.0
	scene_2	2.9	2.0	1.9	2.2	1.9	1.9	3.0
	scene_4	11.4	7.0	7.1	7.5	7.1	7.1	7.8
	scene_6	1.6	1.5	1.5	1.5	1.5	1.5	1.5
	scene_7	✗	12.1	10.6	10.5	11.1	11.2	13.0
	scene_8	3.3	2.8	2.8	2.8	2.8	2.8	2.8
	scene_9	✗	✗	✗	3.1	✗	✗	3.1
	scene_10	3.5	3.1	2.4	2.9	2.4	2.3	2.3
	scene_12	1.4	1.4	1.8	1.6	1.8	1.8	1.6
	scene_13	3.0	2.6	4.2	3.0	4.2	4.2	3.3
	scene_15	3.9	2.4	2.3	2.8	2.3	2.4	3.0
	scene_18	3.5	2.4	2.4	2.6	2.4	2.4	2.7
	scene_19	1.2	1.2	1.1	1.2	1.2	1.2	1.2
	scene_20	2.8	1.7	1.5	2.0	1.5	1.5	2.0
	scene_21	1.6	1.6	1.6	1.7	1.6	1.6	1.7
	scene_29	✗	✗	✗	28.9	✗	✗	3.6
	scene_39	✗	3.8	2.9	13.1	2.9	3.0	3.1
	scene_67	✗	1.6	1.6	1.4	1.9	2.0	1.6
	scene_69	✗	2.2	2.1	2.1	2.1	2.1	2.9
	scene_70	1.8	1.9	1.9	2.2	2.1	1.9	1.9
	scene_106	✗	✗	✗	8.0	✗	✗	2.1
	scene_110	4.2	2.8	1.7	2.2	2.0	1.8	1.8
$N = 4$	scene_25	✗	4.8	4.4	4.2	4.4	4.4	3.9
	scene_30	✗	3.2	3.0	26.5	3.0	3.0	3.1
	scene_31	3.9	4.2	3.6	4.0	3.6	3.6	3.4
	scene_35	3.5	3.5	3.5	3.5	3.5	3.5	3.5
	scene_47	2.1	2.7	2.1	2.4	2.1	2.1	2.2
	scene_54	1.6	1.8	1.4	4.8	1.5	1.5	1.6
	scene_64	1.7	2.5	1.7	5.5	1.8	1.8	1.7
	scene_49	1.4	1.3	1.2	1.5	1.3	1.3	1.4
	scene_92	✗	20.5	✗	30.7	✗	✗	6.3
	scene_81	✗	✗	1.2	20.7	1.7	1.3	1.2
$N = 5$	scene_33	2.5	2.9	2.6	2.7	2.6	2.5	2.6
	scene_46	2.7	3.0	1.7	2.3	3.2	✗	1.6
	scene_51	5.4	2.8	2.6	3.5	2.6	2.6	3.1
	scene_62	4.8	2.1	1.9	2.2	2.1	2.1	2.0
	scene_63	4.0	2.0	1.9	2.0	1.9	1.9	2.1
	scene_65	4.1	2.9	3.3	3.2	3.3	3.3	3.2
	scene_66	✗	2.4	2.3	2.4	2.3	2.3	2.4
	scene_72	1.8	2.1	2.0	2.5	2.2	✗	2.3
	scene_76	✗	3.2	2.8	3.5	2.9	✗	3.5
	scene_77	1.2	1.2	1.1	1.2	1.3	1.2	1.1
	scene_98	1.4	1.2	1.0	✗	✗	1.3	1.3
$N = 6$	scene_27	4.7	3.6	3.4	3.4	3.4	3.3	3.6
	scene_50	✗	✗	2.9	51.6	3.3	✗	3.4
	scene_52	2.4	2.2	2.0	2.3	2.2	2.1	2.4
	scene_53	✗	4.7	4.6	4.7	4.6	4.7	4.8
	scene_55	✗	11.1	3.7	26.4	3.9	3.8	4.0
	scene_57	5.5	2.4	2.2	2.4	2.6	✗	2.4
	scene_58	2.0	2.1	2.5	3.5	2.5	2.4	2.5
	scene_59	2.1	2.1	1.9	2.2	1.9	2.1	2.1
	scene_60	✗	3.8	3.6	4.0	4.0	✗	3.8
	scene_101	✗	✗	✗	47.2	✗	✗	1.4

Table 13: Per-scene Stability score (↑), proposed by [14], the Multi-Camera Benchmark.

	Data	AutoStitch	APAP	GSP	SPW	GES-GSP	OBJ-GSP	MViS(Ours)
$N = 3$	scene_1	✗	.889	.736	.889	.717	.730	.910
	scene_2	.870	.901	.855	.889	.704	.643	.911
	scene_4	.641	.898	.726	.857	.725	.615	.926
	scene_6	.937	.926	.775	.928	.772	.712	.932
	scene_7	✗	.709	.165	.646	.211	.214	.938
	scene_8	.906	.946	.690	.944	.731	.635	.964
	scene_9	✗	✗	✗	.944	✗	✗	.974
	scene_10	.801	.772	.671	.797	.665	.676	.958
	scene_12	.747	.708	.710	.648	.771	.742	.756
	scene_13	.905	.823	.520	.791	.624	.661	.910
	scene_15	.853	.905	.741	.911	.684	.710	.948
	scene_18	.850	.877	.756	.790	.514	.605	.921
	scene_19	.888	.887	.830	.871	.703	.704	.910
	scene_20	.845	.878	.717	.833	.646	.651	.913
	scene_21	.959	.951	.794	.949	.775	.825	.961
	scene_29	✗	✗	✗	.571	✗	✗	.944
	scene_39	✗	.812	.677	.616	.676	.750	.934
	scene_67	✗	.917	.573	.908	.571	.570	.923
	scene_69	✗	.856	.706	.855	.732	.708	.870
	scene_70	.838	.789	.260	.689	.317	.328	.850
	scene_106	✗	✗	✗	.728	✗	✗	.966
	scene_110	.812	.711	.401	.708	.202	.325	.852
$N = 4$	scene_25	✗	.881	.587	.880	.597	.504	.945
	scene_30	✗	.845	.688	.349	.683	.597	.913
	scene_31	.838	.720	.670	.708	.669	.623	.921
	scene_35	.910	.907	.831	.909	.824	.733	.915
	scene_47	.907	.843	.718	.791	.680	.616	.927
	scene_54	.933	.893	.716	.822	.741	.786	.950
	scene_64	.912	.811	.331	.702	.418	.393	.921
	scene_49	.903	.864	.806	.754	.809	.884	.974
	scene_92	✗	.771	✗	.517	✗	✗	.921
	scene_81	✗	✗	.496	.729	.590	.572	.861
$N = 5$	scene_33	.901	.896	.648	.909	.554	.656	.927
	scene_46	.736	.742	.327	.774	.393	✗	.823
	scene_51	.872	.920	.697	.865	.644	.588	.939
	scene_62	.903	.857	.531	.788	.565	.659	.927
	scene_63	.701	.769	.686	.744	.716	.761	.790
	scene_65	.906	.892	.751	.896	.722	.821	.905
	scene_66	✗	.849	.739	.821	.752	.745	.853
	scene_72	.833	.820	.610	.801	.620	✗	.842
	scene_76	✗	.908	.528	.878	.500	✗	.952
	scene_77	.886	.885	.689	.877	.670	.653	.887
	scene_98	.770	.764	.667	✗	✗	.584	.772
$N = 6$	scene_27	.961	.978	.866	.977	.843	.884	.979
	scene_50	✗	✗	.753	.440	.846	✗	.977
	scene_52	.886	.881	.837	.869	.802	.781	.892
	scene_53	✗	.810	.695	.772	.543	.599	.917
	scene_55	✗	.643	.726	.327	.709	.741	.908
	scene_57	.971	.972	.743	.913	.772	✗	.985
	scene_58	.877	.833	.426	.799	.450	.504	.882
	scene_59	.923	.909	.513	.836	.523	.580	.935
	scene_60	✗	.972	.489	.882	.541	✗	.998
	scene_101	✗	✗	✗	.632	✗	✗	.825

Table 14: Per-scene CMSA (↓), the Multi-Camera Benchmark.

	Data	AutoStitch	APAP	GSP	SPW	GES-GSP	OBJ-GSP	MViS(Ours)
$N=3$	scene_1	✗	109	>1k	148	>1k	>1k	0.1
	scene_2	425	109	531	>1k	507	>1k	0.2
	scene_4	>1k	177	>1k	>1k	>1k	>1k	0.1
	scene_6	3.8	97.8	>1k	147	>1k	>1k	0.1
	scene_7	✗	728	>1k	>1k	>1k	>1k	0.1
	scene_8	13.2	13.7	>1k	>1k	>1k	>1k	0.0
	scene_9	✗	✗	✗	>1k	✗	✗	1.3
	scene_10	>1k	>1k	>1k	426	756	751	0.5
	scene_12	1.4	102	>1k	360	822	954	0.4
	scene_13	2.8	113	>1k	204	>1k	755	0.5
	scene_15	151	112	580	>1k	561	303	0.1
	scene_18	142	58.4	288	358	>1k	492	0.1
	scene_19	14.6	13.5	307	561	287	>1k	0.1
	scene_20	21.4	339	318	146	320	495	0.1
	scene_21	1.4	8.9	845	23.7	846	506	0.3
	scene_29	✗	✗	✗	>1k	✗	✗	15.5
	scene_39	✗	>1k	>1k	>1k	>1k	586	20.0
	scene_67	✗	>1k	>1k	>1k	>1k	>1k	2.6
	scene_69	✗	565	>1k	281	>1k	>1k	7.2
	scene_70	70.0	>1k	>1k	>1k	>1k	>1k	3.8
	scene_106	✗	✗	✗	>1k	✗	✗	1.4
	scene_110	31.8	>1k	>1k	571	>1k	>1k	0.1
$N=4$	scene_25	✗	>1k	>1k	>1k	>1k	>1k	13.6
	scene_30	✗	>1k	>1k	>1k	>1k	>1k	5.4
	scene_31	>1k	>1k	>1k	>1k	>1k	>1k	38.7
	scene_35	17.5	48.0	267	45.9	277	>1k	4.4
	scene_47	41.6	>1k	758	>1k	913	>1k	1.1
	scene_54	71.8	417	>1k	>1k	>1k	865	12.2
	scene_64	80.5	>1k	>1k	>1k	>1k	>1k	9.9
	scene_49	17.1	307	>1k	564	>1k	750	0.2
	scene_92	✗	>1k	✗	>1k	✗	✗	18.0
	scene_81	✗	✗	>1k	>1k	>1k	>1k	38.0
$N=5$	scene_33	80.2	110	>1k	107	>1k	>1k	8.2
	scene_46	>1k	>1k	>1k	>1k	>1k	✗	60.9
	scene_51	545	571	>1k	>1k	>1k	>1k	0.3
	scene_62	147	676	>1k	>1k	>1k	>1k	2.5
	scene_63	135	279	>1k	207	982	739	1.0
	scene_65	40.1	109	>1k	107	>1k	>1k	0.3
	scene_66	✗	590	751	>1k	602	562	0.5
	scene_72	20.9	537	>1k	>1k	>1k	✗	6.8
	scene_76	✗	735	>1k	>1k	>1k	✗	13.8
	scene_77	33.7	224	>1k	853	>1k	>1k	26.4
	scene_98	7.5	107	909	✗	✗	>1k	0.6
$N=6$	scene_27	56.0	10.2	>1k	77.8	>1k	>1k	0.2
	scene_50	✗	✗	>1k	>1k	>1k	✗	1.4
	scene_52	26.1	69.1	255	295	202	>1k	0.1
	scene_53	✗	266	542	>1k	>1k	576	0.4
	scene_55	✗	>1k	>1k	>1k	>1k	>1k	0.7
	scene_57	362	285	>1k	>1k	>1k	✗	0.5
	scene_58	15.5	>1k	>1k	>1k	>1k	>1k	3.7
	scene_59	93.9	448	>1k	664	>1k	>1k	0.8
	scene_60	✗	>1k	>1k	>1k	>1k	✗	15.4
	scene_101	✗	✗	✗	>1k	✗	✗	1.2

Table 15: Per-scene GV(↓) comparison on the Multi-Camera Benchmark, **Continued.**

	Data	MIS+TR			MIS+PS			Stab Stitch++	MViS (Ours)
		GSP	GES-GSP	OBJ-GSP	GSP	GES-GSP	OBJ-GSP		
$N=3$	scene_1	2.0	2.0	2.0	1.7	1.7	1.7	6.7	2.0
	scene_2	2.3	2.3	2.3	1.9	1.9	1.9	5.1	3.0
	scene_4	6.7	6.8	6.8	7.1	7.1	7.1	9.1	7.8
	scene_6	1.7	1.7	1.7	1.5	1.5	1.5	5.1	1.5
	scene_7	10.2	10.7	10.9	10.3	10.9	11.0	12.7	13.0
	scene_8	3.0	3.0	3.1	2.8	2.8	2.8	4.6	2.8
	scene_9	✗	✗	✗	✗	✗	✗	9.8	3.1
	scene_10	2.2	2.1	2.1	2.0	2.0	2.0	3.1	2.3
	scene_12	1.9	1.9	1.9	1.7	1.8	1.7	3.3	1.6
	scene_13	4.3	4.3	4.3	4.1	4.0	4.0	4.8	3.3
	scene_15	2.3	2.3	2.6	2.3	2.3	2.4	3.8	3.0
	scene_18	2.4	2.6	2.4	2.3	2.4	2.4	4.1	2.7
	scene_19	1.6	1.9	2.0	1.1	1.1	1.1	15.9	1.2
	scene_20	1.6	1.7	1.6	1.4	1.4	1.4	5.6	2.0
	scene_21	1.9	1.9	1.9	1.6	1.6	1.6	2.2	1.7
	scene_29	✗	✗	✗	✗	✗	✗	118	3.6
	scene_39	3.1	3.1	3.2	3.0	3.0	3.0	32.7	3.1
	scene_67	4.5	5.1	4.9	1.5	1.7	1.7	134	1.6
	scene_69	3.2	3.3	3.2	2.1	2.1	2.1	36.0	2.9
	scene_70	2.6	3.0	2.6	1.8	1.9	1.8	25.5	1.9
	scene_106	✗	✗	✗	✗	✗	✗	166	2.1
	scene_110	1.7	2.3	1.8	1.6	1.7	1.6	4.4	1.8
$N=4$	scene_25	4.5	4.8	4.8	4.2	4.2	4.2	N/A	3.9
	scene_30	3.5	3.5	3.7	3.0	3.0	3.0	N/A	3.1
	scene_31	4.1	4.1	4.0	3.5	3.5	3.5	N/A	3.4
	scene_35	3.6	3.6	3.6	3.5	3.5	3.5	N/A	3.5
	scene_47	2.2	2.2	2.2	2.1	2.1	2.0	N/A	2.2
	scene_54	1.7	1.7	1.8	1.5	1.5	1.5	N/A	1.6
	scene_64	2.2	2.2	2.2	1.6	1.7	1.7	N/A	1.7
	scene_49	1.5	1.7	1.8	1.1	1.3	1.3	N/A	1.4
	scene_92	✗	✗	✗	✗	✗	✗	N/A	6.3
	scene_81	2.0	3.8	2.7	1.1	1.6	1.2	N/A	1.2
$N=5$	scene_33	3.2	3.2	3.2	2.7	2.6	2.6	N/A	2.6
	scene_46	4.4	8.6	5.8	1.5	2.4	✗	N/A	1.6
	scene_51	2.9	2.9	3.0	2.5	2.5	2.5	N/A	3.1
	scene_62	2.3	2.8	2.6	1.8	2.0	2.0	N/A	2.0
	scene_63	2.2	2.2	2.1	1.9	1.9	1.9	N/A	2.1
	scene_65	4.1	4.0	4.2	3.3	3.3	3.4	N/A	3.2
	scene_66	3.1	3.1	3.2	2.3	2.3	2.3	N/A	2.4
	scene_72	3.9	4.5	4.4	1.9	2.0	✗	N/A	2.3
	scene_76	3.7	4.2	4.0	2.8	2.8	✗	N/A	3.5
	scene_77	2.3	2.6	2.8	1.2	1.3	1.3	N/A	1.1
	scene_98	3.4	3.9	3.9	1.0	✗	1.1	N/A	1.3
$N=6$	scene_27	4.2	4.1	4.2	3.4	3.4	3.4	N/A	3.6
	scene_50	3.9	4.6	4.0	2.8	3.2	✗	N/A	3.4
	scene_52	2.3	2.4	2.3	2.0	2.1	2.1	N/A	2.4
	scene_53	4.5	4.5	4.5	4.4	4.4	4.5	N/A	4.8
	scene_55	4.6	4.6	4.6	3.5	3.7	3.6	N/A	4.0
	scene_57	3.3	4.3	3.9	2.2	2.5	✗	N/A	2.4
	scene_58	3.0	3.0	3.0	1.9	1.9	1.9	N/A	2.5
	scene_59	2.1	2.3	2.5	1.8	1.9	2.0	N/A	2.1
	scene_60	3.9	5.3	4.6	3.5	3.9	✗	N/A	3.8
	scene_101	✗	✗	✗	✗	✗	✗	N/A	1.4

Table 16: Per-scene Stab.(↑) comparison on the Multi-Camera Benchmark, **Continued.**

	Data	MIS+TR			MIS+PS			Stab Stitch++	MViS (Ours)
		GSP	GES-GSP	OBJ-GSP	GSP	GES-GSP	OBJ-GSP		
$N = 3$	scene_1	.920	.919	.916	.852	.851	.873	.870	.910
	scene_2	.904	.889	.873	.903	.878	.840	.792	.911
	scene_4	.931	.921	.925	.876	.899	.853	.854	.926
	scene_6	.954	.953	.951	.904	.911	.875	.858	.932
	scene_7	.638	.621	.621	.435	.494	.501	.881	.938
	scene_8	.914	.894	.892	.839	.879	.827	.873	.964
	scene_9	✗	✗	✗	✗	✗	✗	.805	.974
	scene_10	.868	.868	.866	.886	.892	.893	.932	.958
	scene_12	.879	.876	.871	.870	.873	.869	.814	.756
	scene_13	.829	.830	.827	.797	.839	.848	.826	.910
	scene_15	.933	.916	.914	.898	.892	.907	.954	.948
	scene_18	.825	.802	.812	.824	.717	.785	.887	.921
	scene_19	.900	.884	.893	.904	.885	.857	.566	.910
	scene_20	.868	.866	.854	.880	.864	.868	.824	.913
	scene_21	.927	.927	.934	.922	.923	.941	.959	.961
	scene_29	✗	✗	✗	✗	✗	✗	.661	.944
	scene_39	.921	.921	.902	.880	.880	.896	.713	.934
	scene_67	.829	.831	.838	.703	.702	.698	.673	.923
	scene_69	.843	.845	.842	.815	.821	.819	.593	.870
	scene_70	.691	.649	.669	.550	.607	.615	.705	.850
	scene_106	✗	✗	✗	✗	✗	✗	.710	.966
	scene_110	.822	.718	.810	.765	.430	.704	.817	.852
$N = 4$	scene_25	.886	.852	.844	.771	.763	.779	N/A	.945
	scene_30	.953	.952	.915	.923	.923	.827	N/A	.913
	scene_31	.833	.812	.794	.815	.817	.813	N/A	.921
	scene_35	.919	.918	.912	.890	.889	.846	N/A	.915
	scene_47	.966	.954	.941	.898	.864	.864	N/A	.927
	scene_54	.936	.929	.935	.890	.901	.908	N/A	.950
	scene_64	.819	.800	.814	.715	.815	.782	N/A	.921
	scene_49	.961	.957	.949	.938	.952	.960	N/A	.974
	scene_92	✗	✗	✗	✗	✗	✗	N/A	.921
	scene_81	.809	.785	.805	.732	.758	.744	N/A	.861
$N = 5$	scene_33	.933	.923	.912	.841	.779	.884	N/A	.927
	scene_46	.757	.739	.752	.604	.645	✗	N/A	.823
	scene_51	.895	.889	.876	.895	.878	.861	N/A	.939
	scene_62	.828	.804	.822	.828	.859	.878	N/A	.927
	scene_63	.827	.822	.822	.809	.822	.830	N/A	.790
	scene_65	.881	.871	.883	.880	.897	.901	N/A	.905
	scene_66	.829	.821	.815	.802	.820	.817	N/A	.853
	scene_72	.718	.708	.721	.704	.710	✗	N/A	.842
	scene_76	.859	.773	.844	.774	.780	✗	N/A	.952
	scene_77	.761	.762	.743	.775	.771	.741	N/A	.887
	scene_98	.698	.693	.696	.704	✗	.667	N/A	.772
$N = 6$	scene_27	.966	.963	.963	.933	.940	.956	N/A	.979
	scene_50	.957	.939	.933	.925	.960	✗	N/A	.977
	scene_52	.886	.881	.887	.902	.893	.897	N/A	.892
	scene_53	.810	.809	.811	.871	.822	.855	N/A	.917
	scene_55	.911	.902	.897	.915	.913	.920	N/A	.908
	scene_57	.972	.938	.961	.855	.923	✗	N/A	.985
	scene_58	.814	.809	.811	.828	.850	.871	N/A	.882
	scene_59	.757	.749	.754	.780	.801	.804	N/A	.935
	scene_60	.839	.806	.826	.798	.883	✗	N/A	.998
	scene_101	✗	✗	✗	✗	✗	✗	N/A	.825

Table 17: Per-scene CMSA(↓) comparison on the Multi-Camera Benchmark, **Continued.**

	Data	MIS+TR			MIS+PS			Stab Stitch++	MViS (Ours)
		GSP	GES-GSP	OBJ-GSP	GSP	GES-GSP	OBJ-GSP		
$N = 3$	scene_1	4.1	4.0	6.2	47.3	46.9	38.8	1.4	0.1
	scene_2	7.4	7.9	8.7	17.2	17.7	49.1	5.6	0.2
	scene_4	17.8	15.9	16.2	42.0	28.7	59.2	1.3	0.1
	scene_6	2.6	2.7	4.3	93.2	55.0	75.3	1.1	0.1
	scene_7	536	550	533	842	555	427	1.0	0.1
	scene_8	17.1	20.0	94.9	102	35.1	91.4	0.9	0.0
	scene_9	✗	✗	✗	✗	✗	✗	9.4	1.3
	scene_10	14.5	13.9	14.8	35.4	21.9	20.6	0.8	0.5
	scene_12	9.4	9.4	12.9	32.9	38.3	26.7	0.9	0.4
	scene_13	7.4	7.7	8.1	57.5	27.8	19.1	2.2	0.5
	scene_15	5.1	5.2	6.3	22.0	20.5	10.4	0.8	0.1
	scene_18	3.3	6.9	4.8	6.7	48.7	13.9	0.5	0.1
	scene_19	1.5	1.8	108	6.9	6.1	119	45.5	0.1
	scene_20	10.2	10.2	11.4	11.3	10.9	13.3	1.7	0.1
	scene_21	18.6	18.6	18.9	26.2	25.3	14.7	0.7	0.3
	scene_29	✗	✗	✗	✗	✗	✗	36.9	15.5
	scene_39	8.6	8.9	10.0	43.4	42.5	16.8	12.4	20.0
	scene_67	166	163	181	>1k	>1k	>1k	222	2.6
	scene_69	99.5	106	119	404	253	306	35.2	7.2
	scene_70	219	312	220	574	390	383	8.0	3.8
	scene_106	✗	✗	✗	✗	✗	✗	51.1	1.4
	scene_110	47.7	54.1	50.3	102	806	287	0.8	0.1
$N = 4$	scene_25	261	290	278	603	339	340	N/A	13.6
	scene_30	12.4	12.7	135	68.7	68.5	212	N/A	5.4
	scene_31	15.4	15.8	22.3	46.5	45.6	47.7	N/A	38.7
	scene_35	2.9	2.9	10.0	10.4	11.3	41.4	N/A	4.4
	scene_47	4.0	5.1	8.2	19.1	21.7	48.4	N/A	1.1
	scene_54	3.9	4.2	4.1	50.4	30.3	19.9	N/A	12.2
	scene_64	61.6	64.1	66.0	392	178	219	N/A	9.9
	scene_49	11.7	11.8	15.2	47.6	29.4	20.2	N/A	0.2
	scene_92	✗	✗	✗	✗	✗	✗	N/A	18.0
	scene_81	96.9	108	107	>1k	661	819	N/A	38.0
$N = 5$	scene_33	9.0	11.2	15.4	165	193	63.7	N/A	8.2
	scene_46	411	444	447	>1k	>1k	✗	N/A	60.9
	scene_51	31.9	33.4	35.3	81.1	71.8	91.4	N/A	0.3
	scene_62	60.9	62.2	63.9	222	128	119	N/A	2.5
	scene_63	24.0	26.2	32.0	39.6	27.1	20.5	N/A	1.0
	scene_65	23.7	23.8	23.8	111	81.1	63.8	N/A	0.3
	scene_66	4.8	5.3	6.2	32.3	23.0	19.1	N/A	0.5
	scene_72	312	326	332	>1k	>1k	✗	N/A	6.8
	scene_76	126	131	140	224	214	✗	N/A	13.8
	scene_77	17.5	21.3	20.4	168	95.9	86.6	N/A	26.4
	scene_98	8.8	17.2	47.1	29.9	✗	125	N/A	0.6
$N = 6$	scene_27	7.3	9.0	18.0	92.8	59.6	36.8	N/A	0.2
	scene_50	11.1	11.1	18.0	72.9	27.5	✗	N/A	1.4
	scene_52	1.3	1.9	2.8	8.3	6.0	31.7	N/A	0.1
	scene_53	5.3	5.6	5.2	18.1	27.7	15.9	N/A	0.4
	scene_55	24.4	25.8	30.3	67.6	66.1	192	N/A	0.7
	scene_57	12.7	13.3	14.8	467	163	✗	N/A	0.5
	scene_58	84.0	85.9	86.6	222	186	182	N/A	3.7
	scene_59	175	179	179	216	152	147	N/A	0.8
	scene_60	21.5	24.0	23.8	404	129	✗	N/A	15.4
	scene_101	✗	✗	✗	✗	✗	✗	N/A	1.2

Table 18: Summary of notation used in the paper.

Symbol	Description
N	Number of input videos (cameras).
τ	Number of frames in each video.
V_i	i-th input video, $i \in \{1, \dots, N\}$.
F_i^t	Frame t of video V_i.
r	Preprocessing resolution, frames resized to $r \times r$.
$I_i^t \in \mathbb{R}^{r \times r \times 3}$	Resized and RGB-normalized version of F_i^t.
$\boldsymbol{H}_i^t \in \mathrm{SL}(3)$	Predicted homography that aligns I_i^t to the global panorama.
$\widehat{\boldsymbol{H}}_{i \to j}^t$	Observed Relative Transformation (ORT) from I_i^t to I_j^t (same t, different views).
$\widehat{\boldsymbol{H}}_i^{t \to t+1}$	ORT from I_i^t to I_i^{t+1} (same view, consecutive times).
$\widehat{\boldsymbol{H}}_i^{t+1 \to t}$	ORT from I_i^{t+1} to I_i^t.
$M_{i \to j}^t$	Binary mask of the overlap region implied by $\widehat{H}_{i \to j}^t$.
$M_i^{t \to t+1}$, $M_i^{t+1 \to t}$	Temporal overlap masks implied by $\widehat{H}_i^{t \to t+1}$, $\widehat{H}_i^{t+1 \to t}$.
$S_t = \{I_i^s \mid s \in \{t, t+1\},\ i \in \{1, .., N\}\}$	Sliding window at time t.
$\boldsymbol{G} \in \mathbb{R}^{d^2 \times 2}$	Regular $d \times d$ grid in $[-1, 1]^2$ (sampling points for geometric loss).
$\boldsymbol{\theta}_i^t \in \mathbb{R}^8$	Lie-algebra parameters of H_i^t (via the $\mathfrak{sl}(3)$ exponential map).
$\boldsymbol{H}(\boldsymbol{G}) \in \mathbb{R}^{d^2 \times 2}$	The result of applying homography $\boldsymbol{H}$ to $\boldsymbol{G}$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our experiments and results section support the claims. Assumptions are stated in the method section. Limitations are discussed in the Conclusion.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Discussed in the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A].

Justification: Our paper is not theoretical per se, but we do state our assumptions.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Our results are reproducible by following our method. The code is available at: https://github.com/BGU-CS-VIL/MViS

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Most, if not all, information needed for reproducibility is provided either in the paper itself, in the appendix, or in the code. The MMCD dataset is also publicly available. Links to both are on our project page, https://bgu-cs-vil.github.io/MViS-project/.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: There is no training here, just optimization, and we provide all the details.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Due to time and resource constraints, we do not report error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All details are provided.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: There is nothing potentially unethical in our method. Also, the dataset we collected contains no people in it, just scenery.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Partially, yes. We state in the paper that we are unaware of any negative societal impact. We do not discuss potential positive ones, except touching upon possible applications.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification: No safeguard is needed here.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use data from the literature and give due credit of course, as well as data we collected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We constructed a dataset of short videos and described its specifications.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A].

    Justification: Irrelevant for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A].

    Justification: Irrelevant for this work.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A].

    Justification: LLMs are irrelevant for this work.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.