

Eliminating Warping Shakes for Unsupervised Online Video Stitching

Lang Nie^{1,2}, Chunyu Lin^{1,2}*, Kang Liao³, Yun Zhang⁴,
Shuaicheng Liu⁵, Rui Ai⁶, and Yao Zhao^{1,2}

¹ Institute of Information Science, Beijing Jiaotong University

² Beijing Key Laboratory of Advanced Information Science and Network

³ Nanyang Technological University

⁴ Communication University of Zhejiang

⁵ University of Electronic Science and Technology of China

⁶ HAOMO.AI

Abstract. In this paper, we retarget video stitching to an emerging issue, named *warping shake*, when extending image stitching to video stitching. It unveils the temporal instability of warped content in non-overlapping regions, despite image stitching having endeavored to preserve the natural structures. Therefore, in most cases, even if the input videos to be stitched are stable, the stitched video will inevitably cause undesired warping shakes and affect the visual experience. To eliminate the shakes, we propose *StabStitch* to simultaneously realize video stitching and video stabilization in a unified unsupervised learning framework. Starting from the camera paths in video stabilization, we first derive the expression of stitching trajectories in video stitching by elaborately integrating spatial and temporal warps. Then a warp smoothing model is presented to optimize them with a comprehensive consideration regarding content alignment, trajectory smoothness, spatial consistency, and online collaboration. To establish an evaluation benchmark and train the learning framework, we build a video stitching dataset with a rich diversity in camera motions and scenes. Compared with existing stitching solutions, *StabStitch* exhibits significant superiority in scene robustness and inference speed in addition to stitching and stabilization performance, contributing to a robust and real-time online video stitching system. The codes and dataset are available at <https://github.com/nie-lang/StabStitch>.

Keywords: image/video stitching, video stabilization, warping shake

1 Introduction

Video stitching techniques are commonly employed to create panoramic or wide field-of-view (FoV) displays from different viewpoints with limited FoV. Due

* Corresponding author: cylin@bjtu.edu.cn

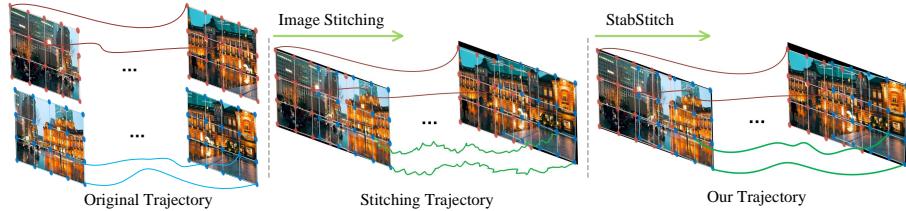


Fig. 1: The occurrence and elimination of warping shakes. Left: stable camera trajectories for input videos. Middle: warping shakes are produced by image stitching, yielding unsMOOTH stitching trajectories. Right: *StabStitch* eliminates these shakes successfully.

to their practicality, they are widely applied in autonomous driving [23], video surveillance [37], virtual reality [38], etc. Our work lies in the most common and challenging case of video stitching with hand-held cameras. It does not require camera poses, motion trajectories, or temporal synchronization. It merges multiple videos, whether from multiple cameras or a single camera capturing multiple videos, to create a more immersive representation of the captured scene.

By contrast, image stitching has been studied much more profoundly, which inevitably throws the question of whether existing image stitching solutions can be extended to video stitching. Pursuing this line of thought, we initially leverage existing image stitching algorithms [15] [45] to process hand-held camera videos. Although the stitched results for individual frames are remarkably natural, there is obvious content jitter in the non-overlapping regions between temporally consecutive frames, as shown in Fig. 1(mid). It is also important to note that the jitter does not originate from the inherent characteristics of the source video itself. In fact, due to the advancements and widespread adoption of video stabilization in both hardware and software nowadays, the source videos obtained from hand-held cameras are typically stable unless deliberately subjected to shaking. For clarity, we define such content jitter as *warping shake*, which describes the temporal instability of non-overlapping regions induced by temporally non-smooth warps, irrespective of the stability of source videos.

Existing video stitching solutions [47] [51] [10] [30] [16] follow a strong assumption that each source video from freely moving hand-held cameras suffers from heavy and independent shakes. Consequently, every source video necessitates stabilization via warping, contradicting the current prevalent reality that video stabilization technology has already been widely integrated into various portable devices (*e.g.*, cellphones, DV cameras, and UAVs). In addition, these approaches, to jointly optimize video stabilization and stitching, often establish a non-linear iterative solving system consisting of various energy terms. Each iteration involves several steps dedicated to optimizing different parameters separately, resulting in a rather slow inference speed and sophisticated optimization.

To solve the above issues, we present the first unsupervised online video stitching framework (termed *StabStitch*) to realize video stitching and video stabilization simultaneously. Building upon the current condition that source videos

are typically stable, we simplify this task to stabilize the warped videos by removing warping shakes as illustrated in Fig. 1 (right). To get stable stitching warps, we generate the stitching trajectories drawing on the experience of camera trajectories (*i.e.*, Meshflow [33]) in video stabilization. By ingeniously combining spatial and temporal warps, we derive the formulation of stitching trajectories in the warped video. Next, a warp smoothing model is presented to simultaneously ensure content alignment, smooth stitching trajectories, preserve spatial consistency, and boost online collaboration. Diverging from conventional offline video stitching approaches that require complete videos as input, *StabStitch* stitches and stabilizes videos with backward frames alone. Besides, its efficient designs further contribute to a real-time online video stitching system with only one frame latency.

As there is no proper dataset readily available, we build a holistic video stitching dataset to train the proposed framework. Moreover, it could serve as a comprehensive benchmark with a rich diversity in camera motions and scenes to evaluate image/video stitching methods. Finally, we summarize our principle contributions as follows:

- We retarget video stitching to an emerging issue, termed *warping shake*, and reveal its occurrence when extending image stitching to video stitching.
- We propose *StabStitch*, the first unsupervised online video stitching solution, with a pioneering step to integrating video stitching and stabilization in a unified learning framework.
- We propose a holistic video stitching dataset with diverse scenes and camera motions. The dataset can work as a benchmark dataset and promote other related research work.

2 Related Work

2.1 Image Stitching

Traditional image stitching methods usually detect keypoints [39] or line segments [54] and then minimize the projective errors to estimate a parameterized warp by aligning these geometric features. To eliminate the parallax misalignment [61], the warp model is extended from global homography transformation [2] to other elastic representations, such as mesh [60], TPS [26], superpixel [24], and triangular facet [25]. Meanwhile, to keep the natural structure of non-overlapping regions, a series of shape-preserving constraints is formulated with the alignment objective. For instance, SPHP [3] and ANAP [29] linearized the homography and slowly changed it to the global similarity to reduce projective distortions; DFW [27], SPW [28], and LPC [15] leveraged line-related consistency to preserve geometric structures; GSP [4] and GES-GSP [6] added a global similarity before stitching multiple images together so that the warp of each image resembles a similar transformation as a whole; etc. Besides, Zhang *et al.* [62] re-formulated image stitching with regular boundaries by simultaneously optimizing alignment and rectangling [12] [44].

Recently, learning-based image stitching solutions emerged. They feed the entire images into the neural network, encouraging the network to directly predict the corresponding parameterized warp model (*e.g.*, homography [42] [46] [17], multi-homography [50], TPS [45] [20] [63], and optical flow [22] [14]). Compared with traditional methods based on sparse geometric features, these learning-based solutions train the network parameters to adaptively capture semantic features by establishing dense pixel-wise optimization objectives. They show better robustness in various cases, especially in the challenging cases where traditional geometric features are few to detect.

2.2 Video Stabilization

Traditional video stabilization can be categorized into 3D [31] [34], 2.5D [32] [7], and 2D [41] [9] [40] methods, according to different motion models. The 3D solutions model the camera motions in 3D space or require extra scene structure for stabilization. The structure is either calculated by structure-from-motion (SfM) [31] or acquired from additional hardware, such as a depth camera [34], a gyroscope sensor [19], or a lightfield camera [49]. Given the intensive computational demands of these 3D solutions, 2.5D approaches relax the full 3D requirement to partial 3D information. To this end, some additional 3D constraints are established, such as subspace projection [32] and epipolar geometry [7]. Compared with them, the 2D methods are more efficient with a series of 2D linear transformations (*e.g.*, affine, homography) as camera motions. To deal with large-parallax scenes, spatially varying motion representations are proposed, such as homography mixture [8], mesh [35], vertex profile [33], optical flow [36], etc. Moreover, some special approaches focus on specific input (*e.g.*, selfie [58] [59], 360 [21] [53], and hyperlapse [18] videos).

In contrast, learning-based video stabilization methods directly regress unstable-to-stable transformation from data. Most of them are trained with stable and unstable video pairs acquired by special hardware in a supervised manner [55] [56] [64]. To relieve data dependence, DIFRINT [5] proposed the first unsupervised solution via neighboring frame interpolation. To get a stable interpolated frame, only stable videos are used to train the network. Different from it, DUT [57] established unsupervised constraints for motion estimation and trajectory smoothing, learning video stabilization by watching unstable videos.

2.3 Video Stitching

Video stitching has received much less attention than image stitching. Early works [16] [48] stitched multiple videos frame-by-frame, and focused on the temporal consistency of stitched frames. But the input videos were captured by cameras fixed on rigs. For hand-held cameras with free and independent motions, there is a significant increase in temporal shakes. To deal with it, videos were first stitched and then stabilized in [11], while [30] did it in an opposite way (*e.g.*, videos were firstly stabilized, and then stitched). Both of them accomplished stitching or stabilization in a separate step. Later, a joint optimization

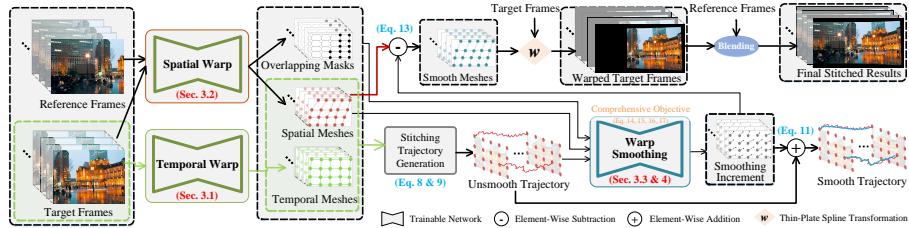


Fig. 2: The overview of *StabStitch*. We first obtain stitching trajectories by integrating spatial and temporal warps. Then the stitching trajectories are optimized by the warp smoothing model to produce unsmooth-to-smooth stitching warps.

strategy was commonly adopted in [51] [10] [47], where [47] further considered the dynamic foreground by background identification. However, solving such a joint optimization problem regarding stitching and stabilization is fragile and computationally expensive. To this end, we rethink the video stitching problem from the perspective of warping shake and propose the first (to our knowledge) unsupervised online solution for hand-held cameras.

3 StabStitch

We first describe the camera trajectories in video stabilization and then further derive the expression of stitching trajectories in video stitching. Afterward, the unsmooth trajectories are optimized to realize both stitching and stabilization. The pipeline of *StabStitch* is exhibited in Fig. 2.

3.1 Camera Trajectory

Temporal Warp: To obtain camera paths, a temporal warp model is first proposed to represent the temporal motion between consecutive video frames. Different from most video stabilization works [57] [35] [33] that use point correspondences to estimate the warp, we leverage a convolutional neural network to capture the high-level information in inter-frame motions. This alternative proves to be robust across various scenarios, particularly in low-light and low-texture environments. The network structure is similar to the warp network of UDIS++ [45]. As shown in Fig. 2(left), it takes two consecutive target frames as input and outputs the motions of mesh-like control points $m(t)$ [1]. Due to the temporal continuity of adjacent frames in the video, the estimated motions are often not significant. Consequently, we replace all global correlation layers [43] in UDIS++ with local correlation layers (*i.e.*, cost volume [52]). To improve the efficiency, we substitute the ResNet50 backbone [13] in UDIS++ with ResNet18 and reduce network parameters accordingly. Following UDIS++, our optimization objective also consists of an alignment term and a distortion

term, as described in the following equation:

$$\mathcal{L}^{tmp} = \mathcal{L}_{alignment} + \lambda^{tmp} \mathcal{L}_{distortion}. \quad (1)$$

The alignment component leverages photometric errors to implicitly supervise control point motions. The distortion component is composed of an inter-grid constraint and an intra-grid constraint. For brevity, we refer the readers to the supplementary for more details.

Meshflow: The camera paths can be defined as a chain of relative motions, such as Euclidean transformations [34], homography transformations [35], etc. Representing the transformation of the initial frame as an identity matrix $F(1)$, the camera trajectories are written as:

$$C(t) = F(1)F(2) \cdots F(t), \quad (2)$$

where $F(t)$ is the relative transformation from the t -th frame to the $(t-1)$ -th frame. Considering that our temporal warp model directly predicts the 2D motions of each control point, we adopt the motion representation of vertex files like MeshFlow [33]. Particularly, we chain the motions of each control point i temporally as the control point trajectory for a more straightforward representation:

$$C_i(t) = m_i(1) + m_i(2) + \cdots + m_i(t), \quad (3)$$

where $m_i(1)$ is set to zero. Note each control point in $m(t)$ is anchored at every vertex in a rigid mesh.

3.2 Stitching Trajectory

Compared with video stabilization, video stitching is more challenging with two or more videos as input and requires the stitched video to possess coherently smooth camera trajectories for the contents from different videos.

Spatial Warp: To obtain the stitching trajectories, in addition to the temporal warp model, we also establish a spatial warp model to represent the spatial motion between different video views, as shown in Fig. 2(left). The spatial warp model has a similar network structure to the temporal warp model except that the first local correlation layer is replaced by a global correlation layer [43] to capture long-range matching (usually longer than half of the image width/height). Considering the significance of spatial warping stability in video stitching, we expect this warp to be as robust as possible, although this network model has been proven to be more robust than traditional methods. To this end, we further introduce a motion consistency term in addition to the basic optimization components of the temporal warp:

$$\mathcal{L}_{consis.} = \frac{1}{(U+1) \times (V+1)} \sum_{i=1}^{(U+1) \times (V+1)} \|m_i(t) - m_i(t-1) - \mu^{spt}\|_2, \quad (4)$$

where μ^{spt} is the maximum tolerant motion difference and $(U + 1) \times (V + 1)$ denotes the number of control points. We further sum up the total optimization goal as:

$$\mathcal{L}^{spt} = \mathcal{L}_{alignment} + \lambda^{spt} \mathcal{L}_{distortion} + \omega^{spt} \mathcal{L}_{consis..} \quad (5)$$

Refer to the ablation studies or supplementary for the impact of $\mathcal{L}_{consis..}$.

Stitch-Meshflow: Video stabilization leverages the chain of temporary motions as camera paths, whereas in our video stitching system, how should we represent the stitching paths of a warped video? We dig into this problem by combining the spatial and temporal warp models. With these two models, we first reach the spatial/temporal motions ($m^S/m^T \in \mathbb{R}^{2 \times (U+1) \times (V+1)}$) and their corresponding meshes ($M^S/M^T \in \mathbb{R}^{2 \times (U+1) \times (V+1)}$) as follows:

$$\begin{aligned} m^T(t) &= TNet(I_{tgt}^{t-1}, I_{tgt}^t) \Rightarrow M^T(t) = M^{Rig} + m^T(t), \\ m^S(t-1) &= SNet(I_{ref}^{t-1}, I_{tgt}^{t-1}) \Rightarrow M^S(t-1) = M^{Rig} + m^S(t-1), \\ m^S(t) &= SNet(I_{ref}^t, I_{tgt}^t) \Rightarrow M^S(t) = M^{Rig} + m^S(t), \end{aligned} \quad (6)$$

where $I_{ref}/I_{tgt} \in \mathbb{R}^{C \times H \times W}$ is the reference/target frame, $SNet/TNet(\cdot, \cdot)$ represents the spatial/temporal warp model, and $M^{Rig} \in \mathbb{R}^{2 \times (U+1) \times (V+1)}$ is defined as the 2D positions of control points in a rigid mesh.

Then we need to derive the stitching motion of the warped video from the spatial/temporal meshes. To align the t -th frame with the $(t - 1)$ -th frame in the warped video, the temporal mesh from the t -th frame to the $(t - 1)$ -th frame in the source video ($M^T(t)$) should also undergo the same transformation as the spatial warp of the $(t - 1)$ -th frame ($M^S(t - 1)$). Assuming $\mathcal{T}(\cdot)$ is the thin-plate spline (TPS) transformation, the desired stitching motion could be represented as the difference between the desired mesh and the actual spatial mesh ($M^S(t)$):

$$s(t) = \mathcal{T}_{M^{Rig} \rightarrow M^S(t-1)}(M^T(t)) - M^S(t). \quad (7)$$

Finally, we attain the stitching paths (we also call it Stitch-Meshflow) by chaining the relative stitching motions between consecutive warped frames as follows:

$$S_i(t) = s_i(1) + s_i(2) + \dots + s_i(t), \quad (8)$$

where we define $s(1)$ is an all-zero array.

3.3 Warp Smoothing

To get a temporally stable warped video, we need to smooth the stitching trajectories as well as preserve their spatial consistency. Besides, we should also try to prevent the degradation of alignment performance in overlapping areas.

Achitecure: In this stage, a warp smoothing model is designed to achieve the above goals. As depicted in Fig. 2, it takes sequences of (N frames) stitching paths (S), spatial meshes (M^S), and overlapping masks (OP) as input, and outputs a smoothing increment (Δ) as described in the following equation:

$$\Delta = SmoothNet(S, M^S, OP), \quad (9)$$

where $S/M^S/OP \in \mathbb{R}^{2 \times N \times (U+1) \times (V+1)}$. OP are binary mask sequences (1/0 indicates the vertex inside/outside overlapping regions). We calculate it by determining whether each control point in M^S exceeds image boundaries.

The smoothing model first embeds S , M^S , and OP into 32, 24, and 8 channels through separate linear projections, respectively. Then these embeddings are concatenated and fed into three 3D convolutional layers to model the spatiotemporal dependencies. Finally, we reproject the hidden results back into 2 channels to get Δ . The network architecture is designed rather simply to accomplish efficient smoothing inference. In addition, this simple architecture better highlights the effectiveness of the proposed unsupervised learning scheme.

With the smoothing increment Δ , we define the smooth stitching paths as:

$$\hat{S} = S + \Delta. \quad (10)$$

Furthermore, if we expand Eq. 10 based on Eq. 8 and Eq. 7, we obtain:

$$\begin{aligned} \hat{S}(t) &= S(t-1) + s(t) + \Delta(t) \\ &= S(t-1) + \mathcal{T}_{M^{Rig} \rightarrow M^S(t-1)}(M^T(t)) - \underbrace{(M^S(t) - \Delta(t))}_{\text{Smooth spatial mesh}}. \end{aligned} \quad (11)$$

In this case, the last term in Eq. 11 can be regarded as the smooth spatial mesh $\hat{M}^S(t)$. Therefore, the sequences of smooth spatial meshes are written as:

$$\hat{M}^S = M^S - \Delta. \quad (12)$$

Objective Function: Given original stitching paths (S) and smooth stitching paths (\hat{S}), smooth spatial meshes (\hat{M}^S), and overlapping masks (OP), we design the unsupervised learning goal as the balance of different optimization components:

$$\mathcal{L}^{smooth} = \mathcal{L}_{data} + \lambda^{smooth} \mathcal{L}_{smoothness} + \omega^{smooth} \mathcal{L}_{space}. \quad (13)$$

Data Term: The data term encourages the smooth paths to be close to the original paths. This constraint alone does not contribute to stabilization. The stabilizing effect of *StabStitch* is realized in conjunction with the data term and the subsequent smoothness term. To maintain the alignment performance of overlapping regions during the smoothing process as much as possible, we further incorporate the awareness of overlapping regions into the data term as follows:

$$\mathcal{L}_{data} = \|(\hat{S} - S)(\alpha OP + 1)\|_2, \quad (14)$$

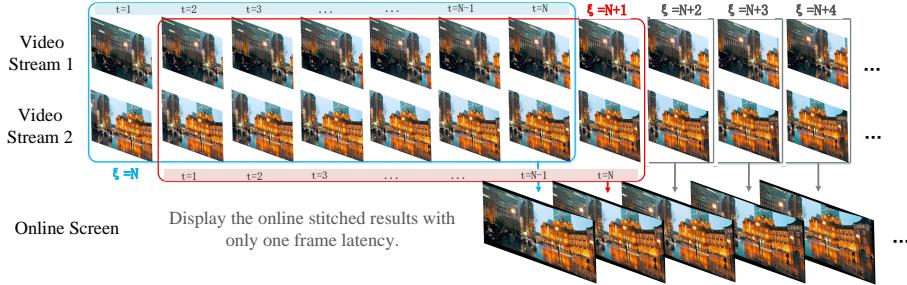


Fig. 3: The online stitching mode. We define a sliding window to process a short sequence and display the last frame on the online screen.

where α is a constant to emphasize the degree of alignment.

Smoothness Term: In a smooth path, each motion should not contain sudden large-angle rotations, and the amplitude of translations should be as consistent as possible. To this end, we constrain the trajectory position at a certain moment to be located at the midpoint between its positions in the preceding and succeeding moments, which implicitly satisfies the above two requirements. Hence, we formulate the smoothness term as:

$$\mathcal{L}_{smoothness} = \sum_{j=1}^{(N-1)/2} \beta_j \|\hat{S}(mid + j) + \hat{S}(mid - j) - 2\hat{S}(mid)\|_2, \quad (15)$$

where mid is the middle index of N (N is required to be an odd number) and β_j is a constant between 0 and 1 to impose varying magnitudes of smoothing constraints on trajectories at different temporal intervals.

Spatial Consistency Term: When there are only data and smoothness constraints, the warping shakes can be already removed. But each trajectory is optimized individually. Actually, our system has $(U+1) \times (V+1)$ control points, which means there are $(U+1) \times (V+1)$ independently optimized trajectories. When these trajectories are changed inconsistently, significant distortions will be produced. To remove the distortions and encourage different paths to share similar changes, we introduce a spatial consistency component as:

$$\mathcal{L}_{space} = \frac{1}{N} \sum_{t=1}^N \mathcal{L}_{distortion}(\hat{M}^S(t)), \quad (16)$$

where $\mathcal{L}_{distortion}(\cdot)$ takes a mesh as input and calculates the distortion loss like the spatial/temporal warp model.

4 Online Stitching

Existing video stitching methods [47] [51] [10] [30] [16] are offline solutions, which smooth the trajectories after the videos are completely captured. Different from

them, *StabStitch* is an online video stitching solution. In our case, the frames after the current frame are no longer available and real-time inference is required.

4.1 Online Smoothing

To achieve this goal, we define a fixed-length sliding window (N frames) to cover previous and current frames, as shown in Fig. 3. Then the local stitching trajectory inside this window is extracted and smoothed according to Sec. 3. Next, the current target frame is re-synthesized using the optimized smooth spatial mesh (Eq. 12). Finally, we blend it with the current reference frame to get a stable stitched frame and display the result when the next frame arrives. With this mode and efficient architectures, *StabStitch* achieves minimal latency with only one frame.

Online Collaboration Term: However, such an online mode could introduce a new issue, wherein the smoothed trajectories in different sliding windows (with partial overlapping sequences) may be inconsistent. This can produce subtle jitter if we chain the sub-trajectories of different windows. Therefore, we design an online collaboration constraint besides the existing optimization goal (Eq. 13):

$$\mathcal{L}_{online} = \frac{1}{N-1} \sum_{t=2}^N \|\hat{S}^{(\xi)}(t) - \hat{S}^{(\xi+1)}(t-1)\|_2, \quad (17)$$

where ξ is the absolute time ranging from N to the last frame of the videos. By contrast, t can be regarded as the relative time in a certain sliding window ranging from 1 to N .

4.2 Offline and Online Inference

Offline smoothing takes the whole trajectories as input, outputs the optimized whole trajectories, and then renders all the video frames. It carries on smoothing after receiving whole input videos and can be regarded as a special online case in which the sliding window covers whole videos. By contrast, online smoothing takes local trajectories as input, outputs the optimized local trajectories, and then renders the last frame in the sliding window. The online process smoothes current paths without using any future frames.

5 Dataset Preparation

We establish a dataset, named *StabStitch-D*, for the comprehensive video stitching evaluation considering the lack of dedicated datasets for this task. Our dataset comprises over 100 video pairs, consisting of over 100,000 images, with each video lasting from approximately 5 seconds to 35 seconds. To holistically reveal the performance of video stitching methods in various scenarios, we categorize videos into four classes based on their content, including regular (RE), low-texture (LT), low-light (LL), and fast-moving (FM) scenes. In the testing



Fig. 4: The proposed *StabStitch-D* dataset with a large diversity in camera motions and scenes. We exhibit several video pairs for each category.

split, 20 video pairs are divided for testing, with 5 videos in each category. Fig. 4 illustrates some examples for each category, where FM is the most challenging case with fast irregular camera movements (rotation or translation). The resolution of each video is resized into 360×480 for efficient training, and in the testing phase, arbitrary resolutions are supported.

6 Experiment

6.1 Details and Metrics

Details: We implement the whole framework in PyTorch with one RTX 4090Ti GPU. The spatial warp, temporal warp, and warp smoothing models are trained separately with the epoch number set to 55, 40, and 50, respectively. λ^{tmp} , λ^{spt} , μ^{spt} , and ω^{spt} are defined as 5, 10, 20, and 0.1. The weights for data, smoothness, spatial consistency, and online collaboration terms are set to 1, 50, 10, and 0.1. α , β_1 , β_2 , and β_3 are set to 10, 0.9, 0.3, and 0.1. The control point resolution and sliding window length are set to $(6 + 1) \times (8 + 1)$ and 7. Moreover, when training the warp smoothing model, we randomly select $N = 7$ frames as the processing window from a larger buffer of 12 frames, which could allow more diverse stitching paths.

Metrics: To quantitatively evaluate the proposed method, we suggest three metrics including **alignment score**, **distortion score**, and **stability score**. Limited by space, we moved the related metric details to the supplementary.

6.2 Compared with State-of-The-Arts

We compare our method with image and video stitching solutions, respectively.

Compared with Image Stitching: Two representative SoTA image stitching methods are selected to compare with our solution: LPC [15] (traditional

Table 1: Quantitative comparisons with image stitching methods on StabStitch-D dataset. * indicates the model is re-trained on the proposed dataset.

	Method	Regular	Low-Light	Low-Texture	Fast-Moving	Average
1	LCP [15]	24.22/0.812	-	-	23.88/0.813	-
2	UDIS++ [45]	23.19/0.785	31.09/0.936	29.98/0.906	21.56/0.756	27.19/0.859
3	UDIS++ * [45]	24.63/0.829	34.26/0.957	32.81/0.920	24.78/0.819	29.78/0.891
4	StabStitch	24.64/0.832	34.51/0.958	33.63/0.927	23.36/0.787	29.89/0.890

Table 2: User study on the cases that Nie *et al.* [47] successes, in which the user preference rate is reported. We exclude the failure cases of Nie *et al.* [47] for fairness.

StabStitch	Nie <i>et al.</i> [47]	No preference
30.47%	6.25%	63.28%

method) and UDIS++ [45] (learning-based method). The quantitative comparison results are illustrated in Tab. 1, where ‘./’ indicates the PSNR/SSIM values. ‘-’ implies the approach fails in this category (*e.g.*, program crash and extremely severe distortion). The results show our solution achieves comparable alignment performance with SoTA image stitching methods. In fact, our spatial warp model has surpassed UDIS++ as indicated in Tab. 4. *StabStitch* sacrifices a little alignment performance to reach better temporally stable sequences.

Compared with Video Stitching: We compare our method with Nie *et al.*’s video stitching solution [47]. To our knowledge, it is the latest and best video stitching method for hand-held cameras. Based on the assumption that the input videos are unstable, it estimates two respective non-linear warps for the reference and target video frame. In contrast, we hold the assumption that currently input videos are typically stable unless deliberately subjected to shaking. Only the target video frame is warped in our system. This difference between Nie *et al.* [47] and our solution makes the comparison of PSNR/SSIM not completely fair. Therefore, we conduct a user study as an alternative and demonstrate extensive stitched videos in our supplementary video.

User Preference: Nie *et al.*’s solution [47] is sensitive to different scenes. In our testing set (20 pairs of videos in total), Nie *et al.* [47] fail in 10 pairs of videos because of program crashes (mainly appearing in the categories of LL and LT). Hence, we exclude these failure cases and conduct the user study only on the successful cases. For a stitched video, different methods may perform differently at different times. So, we segment each complete stitched video into one-second clips (we omit the last clip of a stitched video that is shorter than one second in practice), resulting in 128 clips in total. Then we invite 20 participants, including 10 researchers/students with computer vision backgrounds and 10 volunteers outside this community. In each test session, two clips from different methods



Fig. 5: Qualitative comparison with Nie *et al.*'s video stitching [47] on a regular case (top) and a fast-moving case (bottom). The numbers below the images indicate the time at which the frame appears in the video. Please zoom in for the best view.

Table 3: A comprehensive analysis of inference speed (/ms).

SNet	TNet	Trajectory generation	SmoothNet	Warping	Blending	Total
11.5	10	1.1	1	4.4	0.2	28.2

are presented in a random order, and every volunteer is required to indicate their overall preference for alignment, distortion, and stability. We average the preference rates and exhibit the results in Tab. 2. From that, our results are more preferred by users. Besides, we illustrate two qualitative examples in Fig. 5, where our results show much fewer artifacts (refer to our supplementary video for the complete stitched videos).

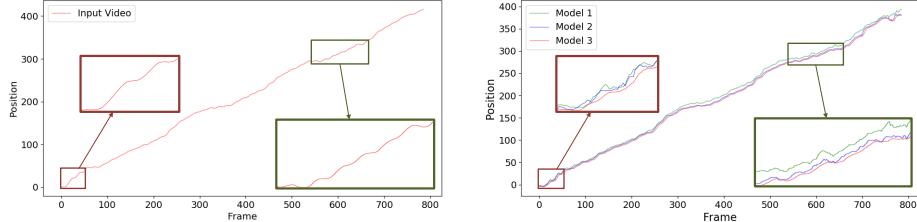
Inference Speed: A comprehensive analysis of our inference speed is shown in Tab. 3 with one RTX 4090Ti GPU, where ‘Blending’ represents the average blending. In the example shown in Fig. 5(top), *StabStitch* only takes about 28.2ms to stitch one frame, yielding a real-time online video stitching system. When stitching a video pair with higher resolution, only the time for warping and blending steps will slightly increase. In contrast, Nie *et al.*'s solution [47] takes over 40 minutes to get such a 26-second stitched video with an Intel i7-9750H 2.60GHz CPU, making it impractical to be applied to online stitching.

6.3 Ablation Study

Quantitative Analysis: The main ablation study is shown in Tab. 4, where ‘Basic Stitching’ (model 1) refers to the spatial warp model without the motion consistency term $\mathcal{L}_{consis.}$. With $\mathcal{L}_{consis.}$ (model 2), the stability is improved. With the warp smoothing model (model 3), both the distortion and stability are significantly optimized at the cost of slight alignment performance, achieving an

Table 4: Ablation studies on alignment, distortion, and stability.

	Basic Stitching	$\mathcal{L}_{consis.}$	Warp Smoothing	Alignment \uparrow	Distortion \downarrow	Stability \downarrow
1	✓			30.67/0.902	0.784	81.57
2	✓	✓		30.75/0.903	0.804	60.32
3	✓	✓	✓	29.89/0.890	0.674	48.74

**Fig. 6:** Left: camera trajectories of the original target video. Right: stitching trajectories of the warped target video from different models (the model index corresponds to the experiment number in Tab. 4).

optimal balance of alignment, distortion, and stability. More experiments can be found in the supplementary materials.

Trajectory Visualization: We visualize the trajectories of the original target video and warped target videos in Fig. 6. Here, the trajectories are extracted from a control point of the example shown in Fig. 5(top) in the vertical direction. It can be observed that even if the input video is stable, image stitching can introduce undesired warping shakes, whereas *StabStitch* (Model 3) minimizes these shakes as much as possible during stitching.

7 Conclusions

Nowadays, the videos captured from hand-held cameras are typically stable due to the advancements and widespread adoption of video stabilization in both hardware and software. Under such circumstances, we retarget video stitching to an emerging issue, *warping shake*, which describes the undesired content instability in non-overlapping regions especially when image stitching technology is directly applied to videos. To solve this problem, we propose the first unsupervised online video stitching framework, *StabStitch*, by generating stitching trajectories and smoothing them. Besides, a video stitching dataset with various camera motions and scenes is built, which we hope can work as a benchmark and promote other related research work. Finally, we conduct extensive experiments to demonstrate our superiority in stitching, stabilization, robustness, and speed.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (No. 62172032), and Zhejiang Province Basic Public Welfare Research Program (No. LGG22F020009).

References

1. Bookstein, F.L.: Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE TPAMI* **11**(6), 567–585 (1989)
2. Brown, M., Lowe, D.G.: Automatic panoramic image stitching using invariant features. *IJCV* **74**(1), 59–73 (2007)
3. Chang, C.H., Sato, Y., Chuang, Y.Y.: Shape-preserving half-projective warps for image stitching. In: *CVPR*. pp. 3254–3261 (2014)
4. Chen, Y.S., Chuang, Y.Y.: Natural image stitching with the global similarity prior. In: *ECCV*. pp. 186–201 (2016)
5. Choi, J., Kweon, I.S.: Deep iterative frame interpolation for full-frame video stabilization. *ACM TOG* **39**(1), 1–9 (2020)
6. Du, P., Ning, J., Cui, J., Huang, S., Wang, X., Wang, J.: Geometric structure preserving warp for natural image stitching. In: *CVPR*. pp. 3688–3696 (2022)
7. Goldstein, A., Fattal, R.: Video stabilization using epipolar geometry. *ACM TOG* **31**(5), 1–10 (2012)
8. Grundmann, M., Kwatra, V., Castro, D., Essa, I.: Calibration-free rolling shutter removal. In: *IEEE International Conference on Computational Photography*. pp. 1–8. IEEE (2012)
9. Grundmann, M., Kwatra, V., Essa, I.: Auto-directed video stabilization with robust l1 optimal camera paths. In: *CVPR*. pp. 225–232. IEEE (2011)
10. Guo, H., Liu, S., He, T., Zhu, S., Zeng, B., Gabbouj, M.: Joint video stitching and stabilization from moving cameras. *IEEE TIP* **25**(11), 5491–5503 (2016)
11. Hamza, A., Hafiz, R., Khan, M.M., Cho, Y., Cha, J.: Stabilization of panoramic videos from mobile multi-camera platforms. *Image and Vision Computing* **37**, 20–30 (2015)
12. He, K., Chang, H., Sun, J.: Rectangling panoramic images via warping. *ACM TOG* **32**(4), 1–10 (2013)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *CVPR*. pp. 770–778 (2016)
14. Jia, Q., Feng, X., Liu, Y., Fan, X., Latecki, L.J.: Learning pixel-wise alignment for unsupervised image stitching. In: *ACM MM*. pp. 1392–1400 (2023)
15. Jia, Q., Li, Z., Fan, X., Zhao, H., Teng, S., Ye, X., Latecki, L.J.: Leveraging line-point consistence to preserve structures for wide parallax image stitching. In: *CVPR*. pp. 12186–12195 (2021)
16. Jiang, W., Gu, J.: Video stitching with spatial-temporal content-preserving warping. In: *CVPRW*. pp. 42–48 (2015)
17. Jiang, Z., Zhang, Z., Fan, X., Liu, R.: Towards all weather and unobstructed multi-spectral image stitching: Algorithm and benchmark. In: *ACM MM*. pp. 3783–3791 (2022)
18. Joshi, N., Kienzle, W., Toelle, M., Uyttendaele, M., Cohen, M.F.: Real-time hyperlapse creation via optimal frame selection. *ACM TOG* **34**(4), 1–9 (2015)
19. Karpenko, A., Jacobs, D., Baek, J., Levoy, M.: Digital video stabilization and rolling shutter correction using gyroscopes. *CSTR* **1**(2), 13 (2011)
20. Kim, M., Lee, Y., Han, W.K., Jin, K.H.: Learning residual elastic warps for image stitching under dirichlet boundary condition. In: *WACV*. pp. 4016–4024 (2024)
21. Kopf, J.: 360 video stabilization. *ACM TOG* **35**(6), 1–9 (2016)
22. Kweon, H., Kim, H., Kang, Y., Yoon, Y., Jeong, W., Yoon, K.J.: Pixel-wise warping for deep image stitching. In: *AAAI*. vol. 37, pp. 1196–1204 (2023)

23. Lai, W.S., Gallo, O., Gu, J., Sun, D., Yang, M.H., Kautz, J.: Video stitching for linear camera arrays. arXiv preprint arXiv:1907.13622 (2019)
24. Lee, K.Y., Sim, J.Y.: Warping residual based image stitching for large parallax. In: CVPR. pp. 8198–8206 (2020)
25. Li, J., Deng, B., Tang, R., Wang, Z., Yan, Y.: Local-adaptive image alignment based on triangular facet approximation. IEEE TIP **29**, 2356–2369 (2019)
26. Li, J., Wang, Z., Lai, S., Zhai, Y., Zhang, M.: Parallax-tolerant image stitching based on robust elastic warping. IEEE TMM **20**(7), 1672–1687 (2017)
27. Li, S., Yuan, L., Sun, J., Quan, L.: Dual-feature warping-based motion model estimation. In: ICCV. pp. 4283–4291 (2015)
28. Liao, T., Li, N.: Single-perspective warps in natural image stitching. IEEE TIP **29**, 724–735 (2019)
29. Lin, C.C., Pankanti, S.U., Natesan Ramamurthy, K., Aravkin, A.Y.: Adaptive as-natural-as-possible image stitching. In: CVPR. pp. 1155–1163 (2015)
30. Lin, K., Liu, S., Cheong, L.F., Zeng, B.: Seamless video stitching from hand-held camera inputs. In: Computer Graphics Forum. vol. 35, pp. 479–487. Wiley Online Library (2016)
31. Liu, F., Gleicher, M., Jin, H., Agarwala, A.: Content-preserving warps for 3d video stabilization. ACM TOG p. 1–9 (2009)
32. Liu, F., Gleicher, M., Wang, J., Jin, H., Agarwala, A.: Subspace video stabilization. ACM TOG **30**(1), 1–10 (2011)
33. Liu, S., Tan, P., Yuan, L., Sun, J., Zeng, B.: Meshflow: Minimum latency online video stabilization. In: ECCV. pp. 800–815. Springer (2016)
34. Liu, S., Wang, Y., Yuan, L., Bu, J., Tan, P., Sun, J.: Video stabilization with a depth camera. In: CVPR. pp. 89–95. IEEE (2012)
35. Liu, S., Yuan, L., Tan, P., Sun, J.: Bundled camera paths for video stabilization. ACM TOG **32**(4), 1–10 (2013)
36. Liu, S., Yuan, L., Tan, P., Sun, J.: Steadyflow: Spatially smooth optical flow for video stabilization. In: CVPR. pp. 4209–4216 (2014)
37. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection—a new baseline. In: CVPR. pp. 6536–6545 (2018)
38. Lo, I.C., Shih, K.T., Chen, H.H.: Efficient and accurate stitching for 360° dual-fisheye images and videos. IEEE TIP **31**, 251–262 (2021)
39. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**, 91–110 (2004)
40. Ma, T., Nie, Y., Zhang, Q., Zhang, Z., Sun, H., Li, G.: Effective video stabilization via joint trajectory smoothing and frame warping. IEEE Transactions on Visualization and Computer Graphics **26**(11), 3163–3176 (2019)
41. Matsushita, Y., Ofek, E., Ge, W., Tang, X., Shum, H.Y.: Full-frame video stabilization with motion inpainting. IEEE TPAMI **28**(7), 1150–1163 (2006)
42. Nie, L., Lin, C., Liao, K., Liu, M., Zhao, Y.: A view-free image stitching network based on global homography. Journal of Visual Communication and Image Representation **73**, 102950 (2020)
43. Nie, L., Lin, C., Liao, K., Liu, S., Zhao, Y.: Depth-aware multi-grid deep homography estimation with contextual correlation. IEEE TCSVT **32**(7), 4460–4472 (2021)
44. Nie, L., Lin, C., Liao, K., Liu, S., Zhao, Y.: Deep rectangling for image stitching: A learning baseline. In: CVPR. pp. 5740–5748 (2022)
45. Nie, L., Lin, C., Liao, K., Liu, S., Zhao, Y.: Parallax-tolerant unsupervised deep image stitching. In: ICCV. pp. 7399–7408 (2023)
46. Nie, L., Lin, C., Liao, K., Zhao, Y.: Learning edge-preserved image stitching from multi-scale deep homography. Neurocomputing **491**, 533–543 (2022)

47. Nie, Y., Su, T., Zhang, Z., Sun, H., Li, G.: Dynamic video stitching via shakiness removing. *IEEE TIP* **27**(1), 164–178 (2017)
48. Perazzi, F., Sorkine-Hornung, A., Zimmer, H., Kaufmann, P., Wang, O., Watson, S., Gross, M.: Panoramic video from unstructured camera arrays. In: *Computer Graphics Forum*. vol. 34, pp. 57–68. Wiley Online Library (2015)
49. Smith, B.M., Zhang, L., Jin, H., Agarwala, A.: Light field video stabilization. In: *ICCV*. pp. 341–348. IEEE (2009)
50. Song, D.Y., Um, G.M., Lee, H.K., Cho, D.: End-to-end image stitching network via multi-homography estimation. *SPL* **28**, 763–767 (2021)
51. Su, T., Nie, Y., Zhang, Z., Sun, H., Li, G.: Video stitching for handheld inputs via combined video stabilization. In: *SIGGRAPH ASIA 2016 Technical Briefs*, pp. 1–4 (2016)
52. Sun, D., Yang, X., Liu, M.Y., Kautz, J.: Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In: *CVPR*. pp. 8934–8943 (2018)
53. Tang, C., Wang, O., Liu, F., Tan, P.: Joint stabilization and direction of 360° videos. *ACM TOG* **38**(2), 1–13 (2019)
54. Von Gioi, R.G., Jakubowicz, J., Morel, J.M., Randall, G.: Lsd: A fast line segment detector with a false detection control. *IEEE TPAMI* **32**(4), 722–732 (2008)
55. Wang, M., Yang, G.Y., Lin, J.K., Zhang, S.H., Shamir, A., Lu, S.P., Hu, S.M.: Deep online video stabilization with multi-grid warping transformation learning. *IEEE TIP* **28**(5), 2283–2292 (2018)
56. Xu, S.Z., Hu, J., Wang, M., Mu, T.J., Hu, S.M.: Deep video stabilization using adversarial networks. In: *Computer Graphics Forum*. vol. 37, pp. 267–276. Wiley Online Library (2018)
57. Xu, Y., Zhang, J., Maybank, S.J., Tao, D.: Dut: Learning video stabilization by simply watching unstable videos. *IEEE TIP* **31**, 4306–4320 (2022)
58. Yu, J., Ramamoorthi, R.: Selfie video stabilization. In: *ECCV*. pp. 551–566 (2018)
59. Yu, J., Ramamoorthi, R., Cheng, K., Sarkis, M., Bi, N.: Real-time selfie video stabilization. In: *CVPR*. pp. 12036–12044 (2021)
60. Zaragoza, J., Chin, T.J., Brown, M.S., Suter, D.: As-projective-as-possible image stitching with moving dlt. In: *CVPR*. pp. 2339–2346 (2013)
61. Zhang, F., Liu, F.: Parallax-tolerant image stitching. In: *CVPR*. pp. 3262–3269 (2014)
62. Zhang, Y., Lai, Y.K., Zhang, F.L.: Content-preserving image stitching with piecewise rectangular boundary constraints. *IEEE TVCG* **27**(7), 3198–3212 (2020)
63. Zhang, Y., Lai, Y., Lang, N., Zhang, F.L., Xu, L.: Recstitchnet: Learning to stitch images with rectangular boundaries. *Computational Visual Media* (2024)
64. Zhang, Z., Liu, Z., Tan, P., Zeng, B., Liu, S.: Minimum latency deep online video stabilization. In: *ICCV*. pp. 23030–23039 (2023)