Real-time Video Stabilization Algorithm based on SuperPoint

Tao Liu[1], Gang Wan[1*], Hongyang Bai[2],Xiaofang Kong[1], Fangyi Wang[1]

***Abstract*—Airborne cameras have become important instruments for measuring and tracking targets, and have a wide range of applications in the field of intelligent unmanned systems for flight patrol. However, the video captured by the camera is subject to external influences and jitter, and the traditional stabilizer is prone to failure in extracting image feature points, while the stabilization effect of the deep learning approach is limited by datasets and the model controllability is weak, making it difficult to achieve real-time. To build a controllable and real-time stabilizer, we make a first attempt to combine traditional methods and deep learning methods and propose a SuperPoint stabilization framework based on deep learning feature point detection in this paper. Firstly, we extract image feature points using SuperPoint neural network, which is better than traditional manual feature point detector. Secondly, homogenize these points. Thirdly, we adopt LK optical flow to improve feature points matching speed and the accuracy for motion estimation. Finally, we define the sliding average filter and the Kalman filter respectively and combine them to smooth unstable trajectories, then compensate motion and output stable video sequences. Experimental results show that our proposed method produces competitive results with current representative methods and more importantly, ours proposed method takes an average of 30ms to stabilize a frame, which is faster than others.**

***Index Terms*—Video Stabilization, Real-Time Process, Trajectory Smoothing**

This work was supported in part by the National Natural Science Foundation of China under Project 62201260 and U2031138, and in part by the National Defense Science and Technology 173 Program Technical Field Foundation of China under Project 2022-JCJQ-JJ-0395.

Tao Liu is with the Nantional Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: liutao14@njust.edu.cn).

Gang Wan is with the Nantional Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: wangang@njust.edu.cn).

Hongyang Bai is with the School of energy and power engineering,Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: hongyang@mail.njust.edu.cn).

Xiaofang Kong is with the Nantional Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: xiaofangkong@njust.edu.cn).

Fangyi Wang is with the Nantional Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China. (e-mail: wangfangyi@njust.edu.cn).

Mentions of supplemental materials and animal/human rights statements can be included here.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) are often equipped with streaming cameras for instant observation[1]. They are popular in several applications such as rescue, surveillance, and mapping. However, the video acquired by them contains unwanted shaking due to the shaking of the airframe itself and atmospheric turbulence, which is not conducive to real-time observation and subsequent target detection, and it needs to be stabilized[2].

Traditional video stabilization methods[3][4][5][6] usually adopt a trajectory-based pipeline consists of three components: 1) global motion estimation, 2) motion smoothing, 3) motion compensation. They work well for small jitter and single scenes, but several challenges degrade these methods. Firstly, the traditional feature points detectors are manually labeled features, and when the texture is sparse, there exist occlusion or large movement, the feature points are poorly represented and even fail to be extracted. Secondly, the extracted feature points are too concentrated and unevenly distributed, it leads to inaccurate estimation movement. Thirdly, the stabilization method for offline video is applied to real-time works, and the performance degrades significantly, and vice versa.

In recent years, some deep learning-based video stabilization methods[7][8][9][10][11][12] have been proposed. Different from traditional methods, deep learning methods either tend to learn the transformation from unstable videos to stable videos or generate stable video sequences directly in an end-to-end manner. Although these methods produce good stabilization results, several issues exist here. First, these methods are less controllable and explainable, and it is difficult to change and improve their network models for complex unstable videos. Second, the training models need to have large-scale stable-unstable video pairs with sufficient variance, but it is not easy to collect these datasets. Third, because deep learning models require high computer computing power, most of these methods are difficult to achieve real time.

To address the aforementioned issues, we combine traditional methods and deep learning methods for the first time, using the excellent feature extraction ability of CNN neural network, combined with the traditional optical flow method and motion smoothing method, and then generate stable video sequences. Specifically, it is divided into four modules: Keypoint Detection, Homogenization Processing, Motion Estimation, and Motion Smoothing.

The main points of work in this paper are threefold:

- Using the neural network to detect the key points of the video frame and homogenizing them, which

increases the accuracy of motion estimation.
- Using the past frames of video, the next moment of the video frame is stabilized, with small computation, high real-time performance, and good stability.
- In the evaluation phase, traditional evaluation metrics are inherited and a new evaluation metric: Retention Ratio is added.

II. Related Work

We briefly introduce the traditional approaches (2D and 3D) and the deep learning-based stabilizer as follows.

A. Traditional Approaches

2D Methods. 2D methods mainly describe the motion trajectory of the camera as a planar two-dimensional motion. One of the most representative ones is the feature point matching method[17]. Key points of all frames in a video sequence are detected and tracked to optimize the camera motion trajectory and output a stable video. Such methods require long-term tracking and matching of key points for motion estimation, but their application scenarios are limited because the appearance around key points may vary significantly, and in addition, manually extracted features are prone to detection failure in occluded and sparsely textured regions. This paper follows this idea, with the difference that the image key points are extracted by neural networks because it has been proven that neural networks have powerful representation capability to increase the effectiveness and stability of keypoint detection.

3D Methods. 3D methods estimate and smooth the camera paths in 3D space and construct point clouds to reconstruct the projections accurately. Liu et al. construct 3D camera paths using motion structure (SFM)[18] and use content-preserving deformation to synthesize stable frames. However, 3D reconstruction of SFM is a challenging task, and estimating camera trajectories is difficult when there are large occlusions in the video frame as well as dynamic objects. In addition, the high requirements of 3D reconstruction for auxiliary hardware also limit application scenarios. In this paper, using 2D methods to stabilize the video sequences captured by the guide head is effective enough.

B. Deep Learning Methods

Deep Learning Methods. Recently, some deep learning-based methods have been applied to video stabilization, but training neural networks requires extensive datasets with sufficient variance. Wang et al. constructed and proposed a DeepStab dataset for stable/unstable videos[7], but the dataset only has 61 stable-unstable video pairs, resulting in insufficient model generalization capability. Huang et al. expanded the DeepStab dataset by adding various instability parameters by computer in 2019, but it still could not simulate the scene depth in the real world[10]. 2020 Choi et al. estimated optical flow through PWCnet[19] network and performed iterative frame interpolation to generate stable videos directly, but the optical flow estimation was time-consuming and ghosting was easily generated around dynamic objects[9]. In the same year, Xu et al. proposed a stabilized video model of DUT[8], whose idea was inspired by the video stabilization of MeshFlow[4] by Liu et al. from the University of Electronic Science and Technology in 2016, and the process was deepened one by one. Although the stabilization of the DUT[8] model is outstanding, it is still not real-time. In 2021, researchers from National Taiwan University and Google jointly introduced a stabilization method that combines the position pose of the gyroscope with neural network training[11], but it requires data from sensors.

By contrast, our method detects key points by the neural network, makes explicit estimation of camera motion, smoothly obtains stable video sequences, has strong model controllability, avoids the problem of insufficient controllability and interpretability of deep learning method models, and does not require a lot of datasets for its training. In comparison with the traditional feature point detection method, it has enough robustness and effectiveness for multiple scenes, good real-time, high stability, and is a pure software algorithm with certain application values.

III. Proposed Method

As shown in Fig. 1, the unstable video is used as input, key point detection, homogenization processing, and camera motion estimation are performed, the obtained trajectory parameters are smoothed, and the motion compensation is performed to output a stable video sequence.

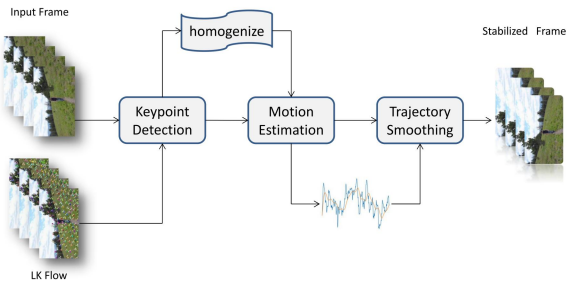


Fig. 1. The pipeline of our video stabilization. Contain four modules: the keypoint detection module, homogenization, motion estimation, and trajectory smoothing module. The keypoint detection module utilizes the detector from SuperPoint[13]. Homogenization utilizes SSC[14]. Optical flow from LK flow[15] is used for motion estimation. The trajectory smoothing module aims at smoothing the estimated trajectories by sliding the average filter and Kalman filter[16].

A. Keypoint Detection

Keypoint detection is the most important part of the Stabilization framework, which directly determines the accuracy of the estimated camera motion. There are two main types of algorithms for extracting feature points and their descriptors: completely hand-designed features and deep learning-based feature extraction.

Completely hand-designed algorithms based on ORB[20], SIFT[21], and SURF[22] algorithms are usually used to extract information mainly by abstracting images through mathematical formulas, whose robustness and generalization capability have

natural disadvantages compared to deep learning algorithms driven by large-scale datasets. Big data-driven deep learning can extract deeper image features than completely hand-designed algorithms, with better robustness and generalization ability, and the end-to-end network for learning features reduces the engineering complexity and also reduces the accumulation of errors.

Feature points and descriptors extracted by deep learning are mainly inferred by convolutional neural networks. The common ones are the L2-Net[23] network proposed by Tian et al., the Hardnet[24] network proposed by Mishchuk et al., and the Key.Net[25] network proposed by Barroso-Laguna et al. and SuperPoint[13] network proposed by DeTone et al. in 2018, which is end-to-end learning of features that input an image and output feature points and descriptors. Among them, only SuperPoint not only is accurate, but has a simple structure that allows for real-time extraction of keypoint. In this paper, we use the SuperPoint keypoint detector

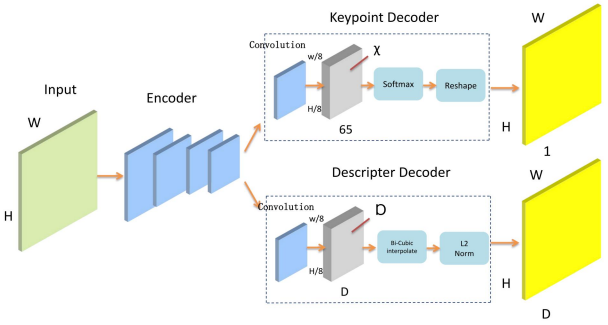


Fig. 2. SuperPoint[13] network, where a full-size image is an input, and a shared encoder similar to the VGG network[26] is used to downsample the image, which is used to extract deep-level image features and reduce the subsequent computation, followed by the feature map size of 1/8 of the original image through the feature point decoder and the descriptor decoder, respectively, and the feature map output from the feature point decoder is convolved by 8 times. The feature map of the same size as the original one is obtained by convolving the sub-pixel of the feature point decoder by 8 times and by upsampling the feature map of the description sub-decoder by 8 times.

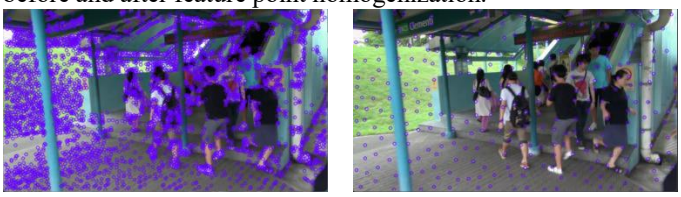
B. Homogenization

After detecting the video frame key points, the camera motion trajectory can be estimated, which is preceded by a step of homogenization. Generally speaking, key points are dense where image edge information is obvious, and sparse where texture is sparse and grayscale changes are flat. Liu et al. MeshFlow[4] to mesh the image to propagate the key point's motion so that the estimated motion is more accurate than before. Mur-Artal et al. use quad trees in ORB-SLAM[27] processing to homogenize key point locations. We use another efficient adaptive non-maximal suppression homogenization processing algorithm with real-time Suppression via Square Covering(SSC)[14].

The SSC algorithm is a process of iterative selection of key points and trial-and-error approximation of a given number. The basic idea is to guess a response distance d, take out the m key points with the highest response and at least d apart in descending order of responsiveness, and reduce the time complexity of the distance calculation to a constant level by using a mask of the same size as the image. Obviously, these m key points must satisfy:

$$r_i = \min \left\| x_i - x_j \right\|, s.t. response(x_j), x_j \in S \quad (1)$$

The critical point response is highest in the region of the range d. If m < n, z, d is reduced by 1/2 of the original size; conversely, d is increased. d is stopped when |m-n| is less than the set threshold k. It is often required to perform 5~8 iterations. The k set in this paper is 10%. Fig. 3 shows the comparison before and after feature point homogenization.



(a) **(b)**

Fig. 3. Comparison of image feature points before and after homogenization. (a) Number and distribution of feature points before homogenization; (b) Number and distribution of feature points after homogenization.

C. Movement Estimation

Our method focuses on estimating the relative motion between the front and back two frames to obtain the original trajectory of the camera. The motion of the two frames before and after the video is divided into translation, rotation, and scaling, and due to the short imaging interval between the two frames before and after the guide head, the effect of scaling on the camera motion can be ignored, so the camera motion can be considered as rigid transformation.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & \Delta x \\ \sin\theta & \cos\theta & \Delta y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (2)$$

Where x, y and x', y' denote the horizontal and vertical coordinates corresponding to a point in the front and back frames, θ denotes the relative rotation angle of the front and back frames around the image center, and Δ x, Δ y denotes the translation components of the corresponding points of the front and back frames in the horizontal and vertical directions. Denote the input video sequence as $\{f_t | \forall t \in [1, E]\}$, where ft is the tth frame of the input video sequence, and E denotes the total number of frames of the input video. Let the original camera path be X_t, and use $H_t(x)$ to denote the motion of the feature points from f_{t-1} to f_t of the adjacent before and after two video pairs of frames, then

$$X_{t+1} = H_{t+1} X_t \quad (3)$$

$$X_t = H_t H_{t-1} \cdots H_1 \quad (4)$$

For the original camera motion path, the stabilized video path

is:

$$P_t = T_t X_t \tag{5}$$

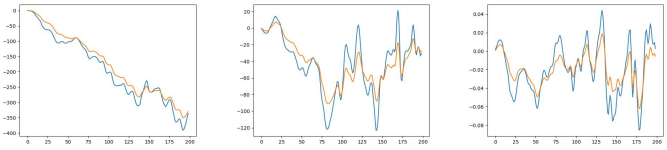
The camera motion H_t is estimated by key points, and then the original camera path is calculated cumulatively, the smoothed path X_t is obtained by smoothing filtering, the video affine transformation matrix required to transform the original path P_t to the smoothed path is calculated by $T_t=P_tX_{t-1}$, and the stable output video sequence is obtained after transforming the original unstable frames.

D. Trajectory Smoothing

The camera motion during the shooting process usually consists of active scanning motion and random dithering motion, which needs to be smoothed by motion smoothing to remove the random dithering motion we do not want and retain the active scanning motion, thus smoothing the camera motion trajectory and improving the video effect of the steady image. In this paper, two types of trajectory smoother are combined to smooth trajectories: moving average filter and Kalman smoothing filter.

Moving Average Filter. A moving average filter is a typical low-pass filter, which can effectively filter the high-frequency noise and retain the useful low-frequency information.

Fig. 4. shows the effect of the moving average filter.



(**a**)x derection path (**a**)y derection path (**c**)a derection path

Fig. 4. Moving average filter. This shows the effect of the moving average filter. The blue curve represents the jitter trajectory of the original video, and the red curve represents the trajectory smoothed by the moving average filter.

Kalman Filter. In real-time applications of natural scenes, the Kalman[16] algorithm is usually applied to smooth the camera trajectory. The Kalman[16] filter algorithm is an algorithm that performs optimal estimation, and the procession consists of two stages: prediction and update.

The prediction procession is as follows:

$$\begin{cases} X(t\,|\,t-1) = \mathrm{F}X(t-1\,|\,t-1) \\ P(t\,|\,t-1) = \mathrm{F}P(t-1\,|\,t-1)\mathrm{F}^T + Q \end{cases} \tag{6}$$

Where X(t|t-1) is the predicted amount of the state corresponding to frame t, X(t-1|t-1) is the optimal estimate of the state corresponding to frame t-1, P(t|t-1) is the state covariance matrix corresponding to frame t, F is the system transfer matrix, and Q is the covariance of the predicted noise.

The update procession is as follows:

$$\begin{cases} X(t\,|\,t) = X(t\,|\,t-1) + M(t)[Z(t) - HX(t\,|\,t-1)] \\ P(t\,|\,t) = (\mathrm{I}\text{-}M(t)H)P(t\,|\,t-1) \\ M(t) = HP(t-1\,|\,t-1)H^T(HP(t\,|\,t-1)H^T + R)^{-1} \end{cases} \tag{7}$$

Where M(t) is the Kalman gain at frame t, Z(t) is the observation matrix, I is the unit matrix, R is the covariance of the measurement noise, R(t|t) is the covariance, and H is the mean square error.

The relevant parameters of our Kalman filter are as follows:

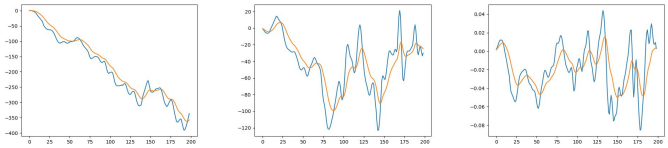
$$X(t=0) = [0,0,0] \tag{8}$$

$$P(t=0) = [1,1,1] \tag{9}$$

$$R = [0.25, 0.25, 0.25] \tag{10}$$

$$Q = [4e-3, 4e-3, 4e-3] \tag{11}$$

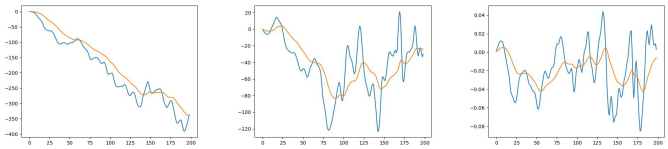
Fig. 5. shows the effect of the Kalman filter.



(**a**)x derection path (**a**)y derection path (**c**)a derection path

Fig. 5. Kalman Smoothing Filter. This shows the effect of the Kalman smoothing filter. The blue curve represents the jitter trajectory of the original video, and the red curve represents the trajectory smoothed by the Kalman filter.

Joint smooth filter. Kalman filter has good real-time performance, but its effect may decline due to inertia in the later stage. Therefore, we use Kalman filter and moving average filter together to better filter out the high frequency part of camera motion. The effect is shown in Fig. 6.



(**a**)x derection path (**a**)y derection path (**c**)a derection path

Fig. 6. joint smoothing filter. The blue curve represents the jitter trajectory of the original video, and the red curve represents the smoothed trajectory.

IV. Experimental Results

We evaluated our model on the public base dataset NUS[5] and compared it quantitatively and qualitatively with other representative methods, including deep learning-based method[7][8][9] and traditional methods[4] in recent years, and conducted an ablation study of the key point detection module. As the subsequent task of stabilized video is target tracking and detection, which requires maximum retention of video information, the extracted stabilized frames are uncropped video frames that have been wrapped by each method.

A. Experiments Settings

We used six types of unstable videos (Regular, Running, Zooming, Crowd, Parallax, and QuickRotating) from NUS[5] to perform the relevant tests. The hardware configuration of the working machine PC on which we conducted the test experiments is as follows: Nvidia Tesla T4 GPU operating system is Windows 10 Professional.

B. Metrics

In this paper, we follow the accepted video quality evaluation metrics PSNR and SSIM, while inheriting distortion and stability from Liu et al.[5] and redefining and introducing the metric-Retention Ratio according to the task.

Fidelity Ratio. Morimoto et al.[28] suggested using the peak signal-to-noise ratio to measure the video fidelity. Intuitively, a stable video sequence is ideally free of excess motion, so there will be no difference among the pixels of two consecutive frames, so the larger the PSNR the more stable the video is. In this paper, the average PSNR of a steady image video is measured as the fidelity of the algorithm.

$$PSNR(I_0, I_1) = 10\log\frac{255^2}{MSE(I_0, I_1)} \tag{12}$$

Where I_0 is the grayscale value of the video frame before stabilization and I_1 is the average of the grayscale after stabilization.

Consistency Ratio. SSIM[29] can often be used to measure image distortion, which is an image quality evaluation method based on structural similarity, by comparing the structure, brightness, and contrast of images to evaluate, and the closer the result is to 1, the more similar the image is before and after stabilization, and the better the algorithm performance, which is used to measure the consistency before and after stabilization in this paper.

$$SSIM(I_0, I_1) = [l(I_0, I_1)]^{\alpha}[c(I_0, I_1)]^{\beta}[s(I_0, I_1)]^{\gamma} \tag{13}$$

Where I_0 is the grayscale value of the video frame before stabilization and I_1 is the average of the grayscale after stabilization. $l(I_0, I_1)$ denotes brightness, $c(I_0, I_1)$ denotes contrast, $s(I_0, I_1)$ denotes structure, α, β, γ are the adjustment parameters for each of the three and are all greater than 0.

Distortion Ratio. Following the criterion proposed by Liu et al.[5], distortion ratio gives the anisotropic scaling of the homograph between an input and output frame. It can be computed by the ratio of the two largest eigenvalues of the affine part of the homograph. Among homograph for all frames, the worst ratio is chosen for the distortion value metric.

Stability Ratio. Following the criterion proposed by Liu et al.[5], stability is mainly used to measure the smoothness of the video. This metric is obtained by extracting the translational and rotational components of the single strain transform between adjacent frames, analyzing them in the frequency domain, allowing two one-dimensional contour signals to be obtained, and calculating the ratio of the lowest energy frequency (usually the 2nd to 6th) to the total energy, taking its minimum value to obtain a final score for stability. The closer this metric is to 1, the more stable the video is.

Retention Ratio. In this paper, the stabilized video sequence is mainly applied to the subsequent target detection and tracking, so it is hoped that the stabilized video will lose as little grayscale information of the original video as possible. The Cropping ratio proposed by Liu et al.[5] is only applicable to evaluate the video methods that perform cropping, but for example, like DIFRINT[9] generates the stabilized video directly by iterative frame interpolation, the cropping ratio is 1, but it does not mean that its stabilized video will not lose the information of the original video, so this paper introduces the concept of retention ratio to evaluate the degree of preservation of video information before and after stabilization. Specifically, the video frames before and after stabilization are grayed out, the average value of the gray information of the video frames before and after stabilization is calculated, and the ratio of the gray value after stabilization to that before stabilization is performed to obtain the retention ratio.

$$Retention = \frac{mean(I_1)}{mean(I_0)} \tag{14}$$

Where I_0 is the grayscale value of the video frame before stabilization and I_1 is the average of the grayscale after stabilization.

C. Quantitative Results

TABLE I

QUANTITATIVE COMPARISON OF DIFFERENT METHODS.THE SCORES ARE AVERAGED OVER EACH CATEGORY. THE HIGHER VALUE OF FIDELITY RATIO, STABILITY RATIO AND DISTORTION RATIO, THE BETTER. THE SUBSCRIPT OF EACH VALUE INDICATES THE RANKING IN THE METHODS COMPARED.

Metrics	Methods	Regular	Running	Zooming	Crowd	Parallax	Quickrotating
Fidelity (PSNR)	Input	22.3487	24.3166	16.2875	21.5119	21.1863	21.7053
	MeshFlow	24.0284(5)	25.3192(4)	18.8334(5)	22.2095(4)	23.2960(3)	21.4966(4)
	DIFRINT	23.5161(3)	25.8947(3)	19.0649(4)	22.8393(3)	22.4004(4)	22.4913(3)
	StabNet	24.3396(4)	20.1562(5)	23.3790(1)	19.2459(5)	20.5422(5)	20.1861(5)
	DUT	24.8750(1)	26.2482(1)	22.2462(2)	24.2708(1)	24.6307(1)	22.4986(2)
	Ours	24.6361(2)	25.9751(2)	20.0054(3)	23.5312(2)	23.4058(2)	23.2548(1)
Consistency (SSIM)	Input	0.4994	0.6552	0.3485	0.6413	0.5464	0.6594
	MeshFlow	0.7919(2)	0.8706(1)	0.5891(3)	0.7928(2)	0.7563(1)	0.7711(2)
	DIFRINT	0.5593(5)	0.7147(5)	0.4864(5)	0.7137(4)	0.6102(4)	0.7391(4)
	StabNet	0.7861(3)	0.7232(4)	0.6905(1)	0.6547(5)	0.5340(5)	0.6744(5)
	DUT	0.7989(1)	0.8588(2)	0.6557(2)	0.8056(1)	0.7425(2)	0.7511(3)
	Ours	0.7826(4)	0.8386(3)	0.5725(4)	0.7701(3)	0.7032(3)	0.7853(1)
Distortion	MeshFlow	0.8878(3)	0.8597(4)	0.9813(2)	0.8248(5)	0.9404(3)	0.9828(1)
	DIFRINT	0.8599(4)	0.9249(3)	0.8603(4)	0.9467(3)	0.7635(4)	0.8604(4)
	StabNet	0.7773(5)	0.6358(5)	0.7806(5)	0.9098(4)	0.4881(5)	0.3267(5)
	DUT	0.9879(1)	0.9710(1)	0.9878(1)	0.9760(1)	0.9850(1)	0.9697(2)
	Ours	0.9749(2)	0.9592(2)	0.9739(3)	0.9689(2)	0.9758(2)	0.8705(3)
Stability	MeshFlow	0.8624(4)	0.8926(2)	0.9450(4)	0.9109(2)	0.8807(3)	0.9367(3)
	DIFRINT	0.8602(5)	0.8319(5)	0.9475(3)	0.9010(3)	0.8604(5)	0.9125(4)
	StabNet	0.8811(3)	0.8457(4)	0.9504(2)	0.8909(4)	0.9057(1)	0.9771(2)
	DUT	0.9059(1)	0.8708(3)	0.9450(4)	0.8596(5)	0.8805(4)	0.9838(1)
	Ours	0.8986(2)	0.8980(1)	0.9608(1)	0.9264(1)	0.8821(2)	0.7612(5)
retention	MeshFlow	0.9562(3)	0.9779(2)	0.9281(4)	0.8462(5)	0.8810(4)	0.8762(4)
	DIFRINT	0.9783(1)	0.9828(1)	0.9849(1)	0.9797(1)	0.9772(1)	0.9813(1)
	StabNet	0.9277(5)	0.9252(5)	0.8482(5)	0.8655(4)	0.8511(5)	0.7757(5)
	DUT	0.9495(4)	0.9438(3)	0.9615(2)	0.9606(2)	0.9653(2)	0.9190(2)
	Ours	0.9578(2)	0.9421(4)	0.9470(3)	0.9469(3)	0.9446(3)	0.9080(3)

Our method was compared with representative methods in recent years, including the traditional method MeshFlow[4], deep learning methods DIFRINT[9], StabNet[7], and DUT[8], and the quantitative results are shown in Table 2. The subscript of each value indicates the ranking in the methods compared. There are several empirical findings:

Firstly, compared with the traditional MeshFlow[4], we use a deep learning approach to extract feature points, while MeshFlow uses a Fast[31] corner point detection algorithm, and the method in this paper outperforms MeshFlow in terms of fidelity, distortion and stability except for the Quickrotating scene. On the one hand, the traditional algorithm extracts feature points quickly but with slightly lower accuracy and

robustness, and even fails in feature point tracking at frame 133 of Quickrotating (the total number of frames is 257), frame 299 of Parallax (the total number of frames is 402), and frame 364 of Crowd (the total number of frames is 977). The SuperPoint[13] network, on the other hand, makes good use of the powerful feature extraction capability of the CNN and therefore demonstrates its superiority in these two most critical metrics. Secondly, for the same real-time method, our method and MeshFlow are comparable in terms of sequence consistency.

Secondly, most scenes perform better than DIFRINT[9] in all five evaluation metrics. The peak signal-to-noise ratio in the Quickrotating scene is lower than that of DIFRINT, although relying on SuperPoint detection to handle more effective key points, it is not enough to compensate for the excessive motion and graphical blur distortion included in the quick rotation to stabilize the video by only compensating for the motion, which is a drawback of almost all trajectory-based methods, but compared with other trajectory-based methods, our method has a slightly higher PSNR[28] value compared to other trajectory-based methods and is better adapted to such motion-complex scenes. More critically, DIFRINT performs video interpolation by iteration, and its stabilization speed is slow and cannot achieve in real-time, while our method is real-time.

Thirdly, DUT[8] is the best-performing method at present. In addition to using CNN to detect image key points, it also uses the famous PWC-net[19] to estimate the motion, and the motion compensation is also adaptive compensation by iterating many times. Therefore, above these five metrics, the combined performance is the best, but the difference between ours, and it is very small, the steady image speed is much faster than DUT, which is because our key point detection network uses SuperPoint[13], while DUTuses RF-net[30], and also LK optical flow[15] tracking is much faster than PWC-net[19]. Our method better achieves the balance of steady image performance and real-time performance.

Fourthly, most scenes perform better than StabNet[7] in these above evaluation metrics. On the one hand, because StabNet implicitly learns the transition from unstable videos to stable videos, it inevitably learns the correct motion incorrectly as redundant jitter, resulting in a poorly controllable and interpretable model. On the other hand, the limited dataset it takes, and the limited scene, is one reason why it fails to exploit the powerful capabilities of CNN. Also, both are real-time methods, ours is less time-consuming than StabNet.

TABLE II
COMPARISON OF STABILIZATION TIME CONSUMPTION BETWEEN METHODS IN DIFFERENT CASES.

Time(s)	MeshFlow	StabNet	DIFRINT	DUT	Ours
Regular	—	0.0969	0.3327	0.1156	0.0247
Running	—	0.0756	0.3282	0.1092	0.0246
Zooming	—	0.1033	0.3422	0.1047	0.0350
Crowd	—	0.0368	0.3268	0.1156	0.0299
Parallax	—	0.0594	0.3373	0.1086	0.0405
QuickRotating	—	0.1028	0.3317	0.1116	0.0275
Average	—	0.0790	0.3330	0.1110	0.0300

Table 2 shows the video stabilization time of each method, where MeshFlow[4] does not expose the python source code. We tried to reproduce the effect based on the paper and could not reproduce the time, so we leave it blank. For DIFRINT[9], it takes 333ms on average to stabilize a frame due to the long iteration process. StabNet[7] has a large variation in time in different scenes due to its supervised learning, which is about 79ms per frame on average. The most outstanding stabilization effect of DUT[8], it takes about 111ms to stabilize one frame, which is faster than DIFRINT[9], but the real-time performance is still insufficient, in addition, the waiting time for loading parameters is long because it contains three modules and four neural networks. In contrast, ours takes only 30ms on average to stabilize one frame, but the stabilization effect is much better than MeshFlow, and the real-time is much better than DUT.

D. Subjective Results

Fig. 7. We present six kinds of videos for subjective evaluation, i.e., 1) Regular, 2) Running, 3) Zooming, 4) Crowd,5) Parallax and 6) QuickRotating in the first row to the sixth row, respectively. It can be observed that our method can handle diverse scenarios compared with traditional stabilizers and deep learning-based methods.

Visualization results of various methods are shown in Fig. 7 (from top to bottom, each row of image scenes is Regular, Running, Zooming, Crowd, Parallax, and Quickrotating in order). In Regular and Quickrotating, MeshFlow[4] may focus too much on dynamic objects because dynamic objects have prominent edge information and more accurate feature point detection, so in sparsely textured lawns and skies, insufficient feature point detection leads to excessive differences in adjacent mesh motion vectors propagated through median filtering, resulting in Wrong clipping and distortion. For the presence of large motion and fast displacement (e.g. the sixth row), the scene information is cropped too much and loses the meaning of the steady image. DIFRINT[9] uses adjacent frame interpolation and is prone to ghosting around dynamic objects or occluded cases, and in the scene where parallax exists in the fifth row, large picture blurring and distortion. StabNet[7] has a general picture effect, and dynamic objects are prone to blurring around them, and in the parallax scene in the fifth row, distortion occurs, which should be caused by learning the normal motion of the video sequence as excess jitter. DUT[8] performs well, but the sparse lawn in the lower right corner is wrong clipping when focused on the third row. ours performance is stable, but for the presence of large motion and large displacement Quickrotating, fewer pixels are stabilized, and some information is lost. In general, ours shows some robustness in different camera motion modes and different

scenes, benefits the powerful representation of CNN feature detection, and also benefits from the powerful tracking capability of LK optical flow[15].

E. Ablation Studies

We conducted an ablation study of the most important feature point detection module in trajectory estimation, replacing the SuperPoint[13] network with conventional ORB[20], SURF[22], and SIFT[21] detectors, respectively, and analyzed the superiority of SuperPoint by visualizing and comparing its feature point matching, as well as the quantitative comparison of the Steadicam performance. After that, the necessity of using the homogenization module is demonstrated, and also shows quantitatively the improvement in video stability performance after homogenization.

Keypoint Ablation Study. Fig. 8. shows a visual comparison of detection and matching of different keypoints. The feature points detected by ORB[20] tend to be concentrated in a certain area of the image, and other texture-sparse areas, such as key points of the sky, lawn, and lake are extremely sparse and there are many cases of mismatching, but their detection speed is extremely fast, usually about 10ms to detect an image. SIFT[21] and SURF[22] compared with ORB, the number of detected feature points and the distribution range rise compared to ORB, but they are prone to false matches. In the second row of the running scene, there are still no valid feature points detected for the sparsely textured sky, while in the fifth row of the inspection, the detected feature points appear redundant. In contrast, the feature points detected by SuperPoint[13] are more widely distributed, and they can be effectively detected for both the sky with sparse texture and edge information and the lawn with slow grayscale changes, and the feature points do not show too much redundancy and mismatching.

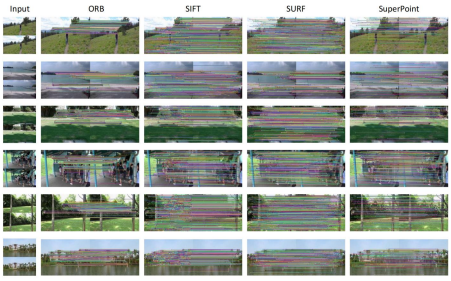


Fig. 8. We show in the first to sixth rows six cases of matching of feature points (different feature point detectors) before and after two frames of video, namely: 1) Regular, 2) Running, 3) Zooming, 4) crowd, 5) parallax and 6) Quickrotation. It can be seen that SuperPoint[13] detects feature points more effectively than the conventional feature point detector.

Homogenization Ablation Study. We tested the effect of homogenization with SSC[14] and without SSC[14] homogenization, as shown in Fig. 9. It can be seen that the steady image performance is improved in all aspects after homogenization.

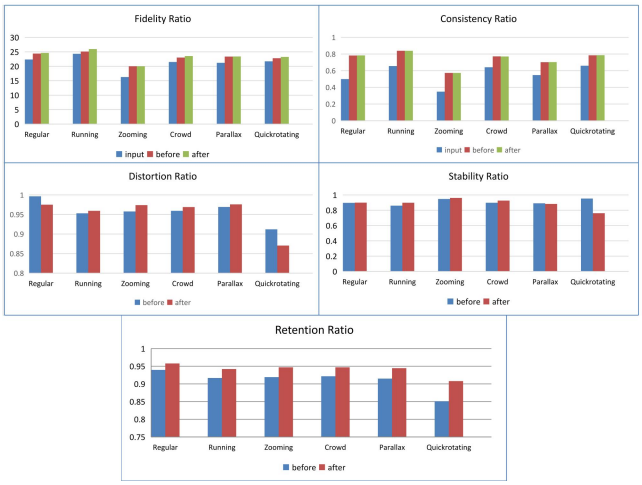


Fig. 9. Quantitative comparison before and after homogenization.

V. Conclusion

We propose a real-time stabilization method based on camera trajectory smoothing, using only past frames to stabilize the next frame. In addition, this paper adds a new stabilization evaluation index Retention Ratio to better measure the retention of video information after stabilization. Our stabilizer, which inherits the advantages of traditional methods and deep learning stabilization methods, is robust to most stabilization scenes, but the missing pixels in the field of view caused by its motion compensation and the black edges left after cropping are still issues to be studied. For the sake of subsequent target detection, this paper cannot be a large crop of video in pursuit of excellent viewing. In recent years, with the technical development of deep learning video enhancement and compensation, we hope to obtain a black-edge processing scheme with minimal loss, which is the next problem to be investigated.

References

[1] A. Lim, B. Ramesh, Y. Yang, C. Xiang, Z. Gao, and F. Lin, "Real-time optical flow-based video stabilization for unmanned aerial vehicles," *J Real-Time Image Proc*, vol. 16, no. 6, pp. 1975–1985, Dec. 2019, doi: 10.1007/s11554-017-0699-y.

[2] "A NOVEL BLOCK MOTION ESTIMATION MODEL FOR VIDEO STABILIZATION APPLICATIONS:," in Proceedings of the Fourth International Conference on Informatics in Control, Automation and Robotics, Angers, France, 2007, pp. 303–306. doi: 10.5220/0001650803030306.

[3] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 1–10, Jan. 2011, doi: 10.1145/1899404.1899408.

[4] B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., "Meshflow: Minimum latency online video stabilization, ". *Computer Vision – ECCV* 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VI, vol. 9910. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-46466-4.

[5] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–10, Jul. 2013, doi: 10.1145/2461912.2461995.

[6] S. Liu, L. Yuan, P. Tan, and J. Sun, "SteadyFlow: Spatially Smooth Optical Flow for Video Stabilization," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, Jun. 2014, pp. 4209–4216. doi: 10.1109/CVPR.2014.536.

[7] M. Wang et al., “Deep Online Video Stabilization With Multi-Grid Warping Transformation Learning,” *IEEE Trans. on Image Process*., vol. 28, no. 5, pp. 2283–2292, May 2019, doi: 10.1109/TIP.2018.2884280.
[8] Y. Xu, J. Zhang, S. J. Maybank, and D. Tao, “DUT: Learning Video Stabilization by Simply Watching Unstable Videos.” arXiv, Jun. 09, 2022. Accessed:Mar.21,2023.[Online].Available:http://arxiv.org/abs/2011.14574
[9] J. Choi and I. S. Kweon, “Deep Iterative Frame Interpolation for Full-frame Video Stabilization,” *ACM Trans. Graph.*, vol. 39, no. 1, pp. 1–9, Feb. 2020, doi: 10.1145/3363550.
[10]C.-H. Huang, H. Yin, Y.-W. Tai, and C.-K. Tang, “StableNet: Semi-Online, Multi-Scale Deep Video Stabilization.” arXiv, Jul. 24, 2019. Accessed:Mar.21,2023.[Online].Available:http://arxiv.org/abs/1907.10283
[11]Y.-L. Liu, W.-S. Lai, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, “Hybrid Neural Fusion for Full-frame Video Stabilization.” arXiv, Aug. 23, 2021. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/2102.06205
[12]M. Zhao and Q. Ling, “PWStableNet: Learning Pixel-Wise Warping Maps for Video Stabilization,” *IEEE Trans. on Image Process.*, vol. 29, pp. 3582–3595, 2020, doi: 10.1109/TIP.2019.2963380.
[13]D. DeTone, T. Malisiewicz, and A. Rabinovich, “SuperPoint: Self-Supervised Interest Point Detection and Description,” in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, Jun. 2018, pp. 337–33712. doi: 10.1109/CVPRW.2018.00060.
[14]O. Bailo, F. Rameau, K. Joo, J. Park, O. Bogdan, and I. S. Kweon, “Efficient adaptive non-maximal suppression algorithms for homogeneous spatial keypoint distribution,” *Pattern Recognition Letters*, vol. 106, pp. 53–60, Apr. 2018, doi: 10.1016/j.patrec.2018.02.020.
[15]L. G. Brown, “A survey of image registration techniques,” *ACM Comput. Surv*.,vol. 24, no. 4, pp. 325–376, Dec. 1992, doi: 10.1145/146370.146374.
[16]Kalman R E . “A New Approach To Linear Filtering and Prediction Problems,” *Journal of Basic Engineering*, 1960, 82D:35-45.
[17]Y. Zhang, H. Yao, P. Xu, R. Ji, X. Sun, and X. Liu, “Video stabilization based on saliency driven SIFT matching and discriminative RANSAC,” in Proceedings of the Third International Conference on Internet Multimedia Computing and Service, Chengdu China, Aug. 2011, pp. 65–69. doi: 10.1145/2043674.2043693.
[18]Ullman S. “The interpretation of structure from motion,”Proc R Soc Lond B Biol Sci. 1979 Jan 15;203(1153):405-26. doi: 10.1098/rspb.1979.0006. PMID: 34162.
[19]D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, “PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume.” arXiv, Jun. 25, 2018. Accessed:Mar.21,2023.[Online].Availablehttp://arxiv.org/abs/1709.02371
[20]E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” in 2011 International Conference on Computer Vision, Barcelona, Spain, Nov. 2011, pp. 2564–2571. doi: 10.1109/ICCV.2011.6126544.
[21]M. Aygun, “Distinctive Image Features from Scale-Invariant Keypoints(SIFT)”.
[22]H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded Up Robust Features,” in Computer Vision – ECCV 2006, vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. doi: 10.1007/11744023_32.
[23]Y. Tian, B. Fan, and F. Wu, “L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, Jul. 2017, pp. 6128–6136. doi: 10.1109/CVPR.2017.649.
[24] A. Mishchuk, D. Mishkin, F. Radenovic, and J. Matas, “Working hard to know your neighbor’s margins: Local descriptor learning loss.” arXiv, Jan. 12,2018.Accessed:Mar.21,2023.[Online].Available:http://arxiv.org/abs/1705.10872
[25]A. Barroso-Laguna, E. Riba, D. Ponsa, and K. Mikolajczyk, “Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters.” arXiv, Oct. 12, 2019. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/1904.00889
[26]K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition.” arXiv, Apr. 10, 2015. Accessed: Mar. 21, 2023. [Online]. Available: http://arxiv.org/abs/1409.1556
[27]R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras,” IEEE Trans. Robot., vol. 33, no. 5, pp.1255–1262, Oct. 2017, doi: 10.1109/TRO.2017.2705103.
[28]H. Yao, M.-Y. Huseh, G. Yao, and Y. Liu, “Image Evaluation Factors,” in Image Analysis and Recognition, vol. 3656, M. Kamel and A. Campilho, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 255–262. doi: 10.1007/11559573_32.
[29]Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image Quality Assessment: From Error Visibility to Structural Similarity,” IEEE Trans. on Image Process., vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[30]X. Shen et al., “RF-Net: An End-To-End Image Matching Network Based on Receptive Field,” in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, Jun. 2019, pp. 8124–8132. doi: 10.1109/CVPR.2019.00832.
[31]E. Rosten and T. Drummond, “Machine Learning for High-Speed Corner Detection,” in Computer Vision – ECCV 2006, vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 430–443. doi: 10.1007/11744023_34.

Tao Liu is currently pursuing the master's degree with Nanjing University of Science and Technology. His research interests include image processing and computer vision.

Gang Wan received the Ph.D. degree from Nanjing University of Science and Technology, China, in 2013. He is currently an Associate Professor with the National Key Laboratory of Transient Physics, Nanjing University of Science and Technology. His research interests include image processing, weapon effectiveness evaluation, ballistic test technology, and engine control technology.

Hongyang Bai received the Ph.D. degree from Nanjing University of Science and Technology, China, in 2012. He is currently a Professor with the School of energy and power engineering, Nanjing University of Science and Technology. His research interests include image processing, guidance and navigation and engine control technology.

Xiaofang Kong is currently an Associate Professor with the National Key Laboratory of Transient Physics, Nanjing University of Science and Technology. His research interests include image processing and computer vision.

Fangyi Wang is currently pursuing the master's degree with Nanjing University of Science and Technology. His research interests include computer vision.