Digital Video Stabilization: Methods, Datasets, and Evaluation

Marcos Roberto e Souza[1], **Helena de Almeida Maia**[1], **Hélio Pedrini**[1]

[1]Institute of Computing – State University of Campinas (UNICAMP)
Av. Albert Einstein, 1251, 138083-852 – Campinas – SP – Brazil

{marcos.souza,helena.maia,helio}@ic.unicamp.br

Abstract. *Video stabilization removes shaky camera motion from videos. In our thesis, we presented an extensive review, including a formal problem definition, meta-analysis, and other elements, resulting in two survey papers. We introduced new measures for stability assessment and studied the correlation between them and human perception. We also proposed a novel evaluation approach for 2D camera motion estimation. We then introduced NAFT, a semi-online DWS method with a neighborhood-aware mechanism to stabilize without an explicit stability definition. We supervised NAFT with SynthStab, our proposed synthetic dataset. NAFT closed the quality gap with non-DWS methods while reducing the number of parameters and model size by 14×.*

1. Introduction

The goal of video stabilization is to obtain a stabilized video by changing the camera motion of a shaky video. This can be done entirely through software (digital video stabilization), eliminating the need for hardware stabilizers and making it a cost-effective solution. Additionally, it is the only option for improving videos that have already been recorded.

Different approaches to digital video stabilization are categorized by how they utilize video data. *Online stabilization* analyzes only the current and preceding frames to make adjustments. This is proper for situations where you need to stabilize videos during recording, such as live streaming. In contrast, *offline stabilization* analyzes the entire video simultaneously, allowing for a more comprehensive understanding of motion patterns, typically resulting in superior stabilization quality. However, this method requires access to the entire video beforehand and would not work for some applications. Finally, *semi-online stabilization* bridges the gap between these two. It relaxes the offline stabilization by processing a set of frames at once. This allows for better utilization of the available data while still enabling some online-like processing. Offline, online, and semi-online methods can operate either in real-time or not, representing a distinct classification.

Video stabilization traditionally relies on a three-step process to smooth out shaky footage: camera motion estimation, unwanted motion determination and stabilized view rendering. The first stage involves calculating the camera's path during recording. Next, unwanted shakes are identified and removed from this path, resulting in a smoother trajectory. Finally, the video frames are repositioned based on the refined camera path, creating the final stabilized video. Recently, a new approach called direct warping stabilization (DWS) has emerged, which directly predicts the necessary transformation to stabilize each shaky frame. This essentially combines the steps of motion estimation and

unwanted motion determination into a single process. Researchers argue that DWS performs better on low-quality videos and requires fewer computational resources compared to traditional methods [Zhao and Ling 2020].

Our research delves into both traditional and DWS approaches to video stabilization. For traditional methods, we advocate for a deeper understanding by analyzing each step independently. To achieve this, we propose a new evaluation method specifically focused on the motion estimation stage. For DWS methods, we developed a new stabilization technique that adheres to this proposed framework. Our research aimed to identify and address critical weaknesses in current video stabilization research, while also proposing new techniques and methodologies. Due to time constraints, we prioritized addressing three key issues: (i) the lack of well-organized literature, for which we provided a comprehensive review and organization of existing research on video stabilization; (ii) inadequate rigor in assessments and limited knowledge about the effectiveness of the metrics, for which we discussed and expanded the current knowledge regarding stabilization assessment; and (iii) the fact that DWS does not achieve the same stability quality as traditional approaches, so we improved the effectiveness and efficiency of DWS methods.

The thesis yielded the following main products: (i) a critical and detailed review of digital video stabilization methods (first survey) [Souza et al. 2022]; (ii) a critical and detailed review of video stabilization assessment and datasets (second survey) [Souza et al. 2023c]; (iii) a metric for assessing the two-dimensional camera motion estimation (Section 2) [Souza et al. 2023b]; (iv) new evaluation measures for the final stabilization quality based on the kinematics of pixel profiles (Section 2); (v) a new synthetic dataset with paired stable and unstable videos (Section 3) [Souza et al. 2023a]; and (vi) a new direct warping stabilization method (Section 4) [Souza et al. 2023a]. Because of the textual nature and length of the first two products, they can be consulted in the thesis but are not included in this summary.

2. Video Stabilization Assessment

Figure 1 shows the steps that could be assessed following the classical stabilization approach. Most literature works only perform the final stabilization assessment (Step 4). We conjecture that each step must be followed by a specific assessment, conducted as independently as possible, besides a final overall evaluation. Besides, each step could have its own datasets with proper ground-truth data. A possible approach is to evaluate automatically the first three steps according to relevant physical properties. In our ideal scenario, this final assessment may play an additional role by guiding the physical-based measures from Steps 2 and 3. The end user is typically a human, so the final evaluation should focus on human perception. In this work, we only proposed an assessment strategy for the motion estimation step, presented as follows.

2.1. Rethinking 2D Camera Motion Assessment

While significant advancements have been made in camera motion estimation, there is a lack of thorough evaluation of 2D methods, which are crucial for traditional video stabilization. To bridge this gap, we introduce a novel evaluation method using a pixel-by-pixel comparison of camera motion fields. Our experiments demonstrate the robustness of our metrics across various situations, outperforming conventional image similarity metrics.

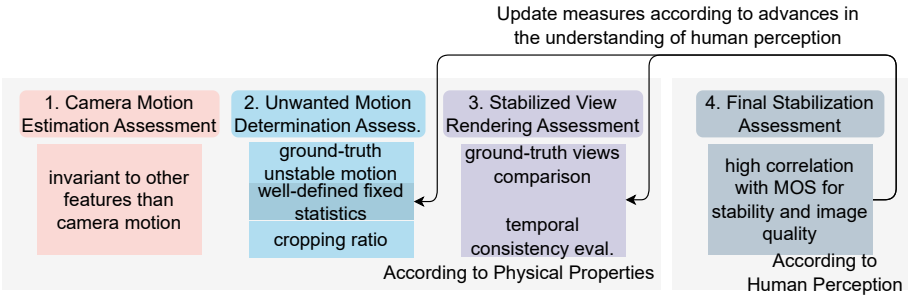
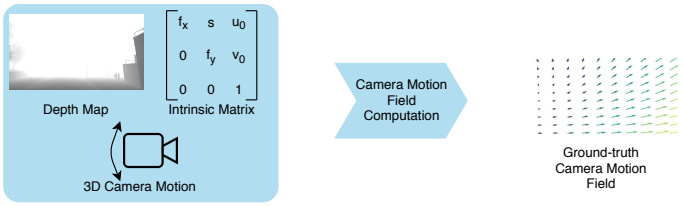
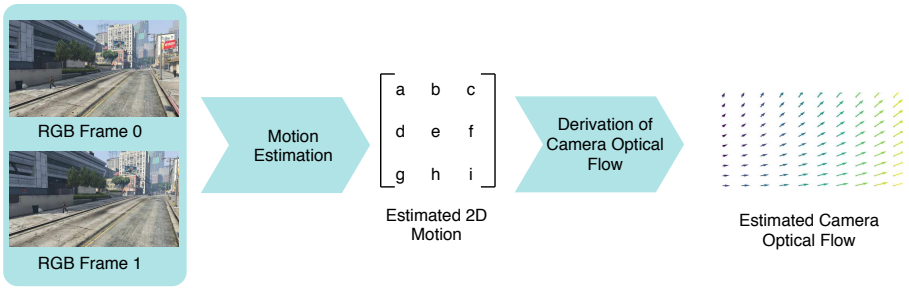


Figure 1. Diagram of our ideal process proposed for the video stabilization assessment. Source: Author's Thesis.

As presented in Figure 2, our evaluation process involves several steps. First, we establish the ground-truth camera motion field using: (i) the relative 3D motion between frames in the video, (ii) the depth information, and (iii) the camera's intrinsic parameters (refer to Subfigure 2a). It is important to note that this evaluation method requires datasets with this comprehensive information. In contrast, the method being evaluated only utilizes RGB frames as input (refer to Subfigure 2b). In this case, we calculate the camera optical flow from the estimated motion. Finally, we perform a pixel-by-pixel comparison using established metrics from the field of optical flow analysis. We choose the representation with the highest level of detail to enable a comprehensive comparison across different representations and degrees of freedom.



(a) ground-truth information



(b) 2D motion estimation

Figure 2. Main steps of the two-dimensional motion estimation assessment method. Source: Author's Thesis.

2.1.1. Experimental Results

In Table 1, we show the relationship between established image similarity metrics (PSNR and SSIM) and our proposed metrics (AEPE and Fl), as detailed in Chapters 2 and 5

of the thesis. Due to the opposing interpretations of these metrics (higher similarity vs. lower error for EPE), we aimed for a correlation of approximately -1. This analysis is particularly relevant since the few works that assess 2D camera motion use similarity measures.

Table 1. Correlation between image similarity and EPE-based metrics for test splits.

Dataset	Image Similarity	AEPE Average		Fl	
		PLCC	SROCC	PLCC	SROCC
TartanAir	**PSNR**	-0.2789±0.29	-0.6350±0.19	-0.5535±0.16	-0.5439±0.17
	SSIM	-0.2918±0.29	-0.7032±0.13	-0.6990±0.18	-0.6935±0.18
MVS-Synth	**PSNR**	-0.2390±0.01	-0.4286±0.01	-0.3528±0.01	-0.4060±0.01
	SSIM	-0.3605±0.06	-0.5886±0.03	-0.6571±0.03	-0.6311±0.02
KITTI	**PSNR**	-0.2598±0.19	-0.6609±0.23	-0.5740±0.26	-0.5627±0.28
	SSIM	-0.2847±0.19	-0.7272±0.26	-0.6238±0.28	-0.5694±0.30

Our findings revealed a weak correlation between traditional similarity metrics and our proposed approach. To understand this, we analyzed specific cases where EPE-based and similarity metrics diverged. Table 2 delves into these scenarios, highlighting situations where similarity metrics failed while our method succeeded. Our approach consistently demonstrated accurate performance in these discrepancies, likely due to its superior ability to isolate motion, unlike similarity metrics which can be influenced by various factors.

Table 2. Description of main cases where similarity metrics do not seem to be adequate to assess the quality of camera motion estimation.

Case	Description	Expected Behavior
Low-textured Frames	Frames where neighboring pixels are very similar.	Similarity metrics do not show much difference when we change the camera motion.
High-textured Frames	Frames where neighboring pixels are very different.	Similarity metrics can be very distinct, even with low changes in the camera motion.
Abrupt Camera Motion	Relative camera motion for two frames is very large.	Borders generated on warped images significantly reduce the similarity value of images.
Large Moving Objects	Many pixels are covered by moving objects.	Compensating for camera motion results in low similarity in pixels of moving objects.
Lighting Variation	Pixels are affected by a change in lighting.	Low values in similarity metrics in regions affected by lighting variation.

A key limitation of our method is its dependence on controlled datasets. However, we argue that this limitation is acceptable for rigorous quality assessment, especially with access to a large amount of high-quality data. Additionally, it is important to acknowledge challenges with reflective surfaces, where camera motion effects differ from those on non-reflective surfaces.

2.2. An Analysis on Final Stability Assessment

In this section, we introduce novel kinematic measures using the first, second, and third derivatives from pixel profiles [Liu et al. 2014]. The inspiration for investigating these measures originated from visual observations: we found that, in simple videos, techniques based on kinematic principles [Grundmann et al. 2011] outcome videos with better stability than those relying on high-frequency attenuation [Liu et al. 2013]. Later, we used

these measures and other statistics to compose a feature vector and train a regressor to predict stability scores based on human perception.

We named our three kinematic measures from the pixel profiles as Velocity of Camera Pixel Profiles (VCP^2), Acceleration of Camera Pixel Profiles (ACP^2) and Jerk of Camera Pixel Profiles (JCP^2). For them, we used a segmentation mask between static regions and moving objects to ignore the latter. The utilization of pixel profiles instead of feature trajectories was driven by two primary reasons: (i) pixel profiles offer a dense representation that encompasses all pixels within each frame, and (ii) their implementation is more straightforward, as we do not need to track features across the entire video, avoiding complications such as temporal discontinuity and features leaving the image domain.

Later, we hypothesized that stability assessment should encompass multiple aspects. We defined the aspects based on existing methodologies in the literature: (i) image similarity, (ii) frequency analysis, (iii) the geometry, and (iv) the kinematic measures. We defined distinct measures for each aspect. Typically, these measures output a value per pixel or frame. In this way, we used six statistical metrics to synthesize these values: average, standard deviation, median, interquartile range, kurtosis, and skewness.

2.2.1. Experimental Results

Table 3 presents the correlations between the human perception stability scores (LIVE-Qualcomm and MIND-VQ datasets), as well as (i) different methods of evaluating the stability from the literature, (ii) the kinematic measures, and (iii) different regression methods that use multiple measures as input. We reported the averages of the correlation values across the 10 test subsets (even for untrainable measures). We experimented with five regressors as well as a Linear Fit, where we fit a simple straight line on the features. For each regressor, we perform a 3-fold cross-validation for a hyperparameter grid search.

Table 3. Correlation between human perception of stability scores and different strategies for assessing stability.

Measure	LIVE-Qualcomm		MIND-VQ	
	PLCC	SROCC	PLCC	SROCC
LHR	0.388	0.404	0.538	0.489
ITF (PSNR)	0.015	0.024	0.317	0.308
ITF (SSIM)	0.081	0.072	0.200	0.196
IGC	0.626	0.614	-	-
VCP^2	0.204	0.405	0.546	0.555
ACP^2	**0.710**	0.720	**0.781**	**0.769**
JCP^2	0.636	**0.755**	0.753	0.747
Linear Fit	0.142	0.181	0.706	0.767
SVR	-	-	0.853	0.815
RF	-	-	0.880	0.839
GBM	-	-	**0.884**	**0.843**
XGBoost	-	-	**0.884**	0.842

In most instances, ACP^2 exhibited the highest correlation values among the stability measures, but for SROCC in the LIVE-Qualcomm dataset. The commonly used metrics (LHR and ITF) achieved low correlations. These outcomes with our meta-analysis

suggest that the quantitative results reported in the literature (typically relying on LHR) not only demonstrate limited consensus with each other but also lack robust alignment with human perceptions of stability. The other results refer to trained regression models to predict the stability score from the features of multiple aspects. As LIVE-Qualcomm has few videos, it easily overfits, even fitting a straight line in the data (Linear Fit). Concerning MIND-VQ, we increased the correlation values by 10.3 percentage points compared to the highest-performing non-machine learning-based evaluation measure. Nevertheless, this approach exhibited limited robustness in inter-dataset experiments.

3. SynthStab

This section presents SynthStab, a novel synthetic dataset featuring paired videos designed for training models using camera motion as supervision, rather than solely relying on pixel-level similarity. We achieve this by leveraging the principles of kinematics to generate realistic camera movements. We utilize the environments available within Unreal Engine, and AirSim to create videos showcasing a spectrum of realistic and dynamic scenarios (Figure 3). We retain control over camera motion throughout the process. All frames are rendered at a resolution of 512×256 pixels. For each pair of stable and unstable videos, we provide RGB frames, dense depth maps, and motion fields between each frame pair. With over 102,400 frames, SynthStab provides a substantial pool of data to train deep learning models, a critical component for DWS methods. We randomly partition our dataset into training and validation sets. Figure 4 outlines the process of constructing SynthStab.

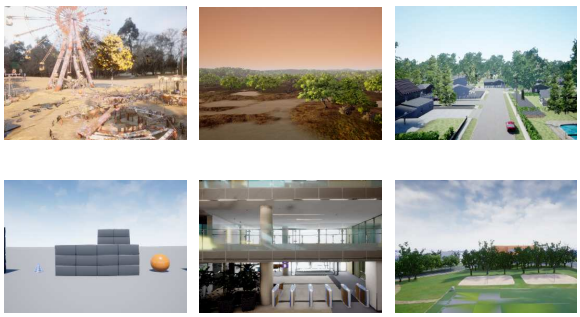


Figure 3. Environments present in our dataset. We have simple, complex, indoor and outdoor environments. Source: Author's Thesis

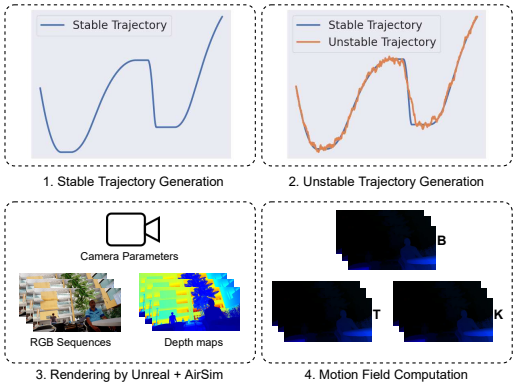


Figure 4. Overview of the construction process of our dataset. Source: Author's Thesis.

The core principle involves generating stable camera trajectories that encompass the six degrees of freedom (6-DoF) of 3D camera movement. We calculate each of these six variables independently, with trajectory segments categorized into constant position segments (CPS), constant velocity segments (CVS), and constant acceleration segments (CAS), similar to the kinematics concepts used by Grundmann et al. 2011. Next, we define unstable trajectories by introducing random keypoints and establishing a random path between them, all while staying consistent with the core elements of the stable trajectory. After generating a predetermined number of trajectories, we render video pairs for each trajectory and environment using Unreal Engine and AirSim. Finally, we compute dense camera motion fields by performing an inverse projection process, utilizing the

stable frame's depth map, the relative 3D motion matrix between frames, and the intrinsic camera parameters.

4. NAFT

We introduce Neighborhood-aware recurrent All-pairs Field Transforms (NAFT), a novel technique for video stabilization that leverages direct warping in a semi-online manner. NAFT adapts the RAFT algorithm for this purpose and incorporates a neighborhood-aware update mechanism called IUNO. Through a training process on SynthStab data combined with IUNO, the model learns to identify characteristics of video stability directly from the data patterns, without relying on predetermined stability definitions. Furthermore, we demonstrate how a pre-existing video inpainting method can be combined with NAFT to achieve full-frame stabilization. Our experiments show that NAFT achieves superior stabilization performance even under significant camera motion, outperforming other direct warping methods and approaching the state-of-the-art. Notably, our smallest network variant (NAFT-S) requires only around 7% of the model size and trainable parameters compared to the smallest model among competing methods.

4.1. Proposed Method

Figure 5 illustrates our training pipeline. We denote a sequence of unstable RGB frames as $\mathbf{V}_i = \{\mathbf{F}_{i-d_\Omega}, \mathbf{F}_{i-d_{\Omega-1}}, \cdots, \mathbf{F}_i, \cdots, \mathbf{F}_{i+d_{\Omega-1}}, \mathbf{F}_{i+d_\Omega}\}$, where each frame $\mathbf{F}_\omega \in [0,1]^{H\times W\times 3}$ and $\mathbf{d} = \{d_1, \cdots, d_{\Omega-1}, d_\Omega\}$ represents the displacements of the input sequences. Similarly, we define $\mathbf{M}_i^{\text{ngb}} = \{\mathbf{M}_{i-d_\Omega}, \mathbf{M}_{i-d_{\Omega-1}}, \cdots, \mathbf{M}_{i+d_{\Omega-1}}, \mathbf{M}_{i+d_\Omega}\}$ as a sequence of motion fields corresponding to neighboring frames, where $\mathbf{M}_\omega \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 2}$ warps its corresponding unstable frame $\mathbf{F}_\omega$ into a stable version $\bar{\mathbf{F}}_\omega$. Given $\mathbf{V}_i$ and $\mathbf{M}_i^{\text{ngb}}$, our objective is to predict the optical flow $\mathbf{B}_i$ of size H×W×2, which transforms the unstable frame $\mathbf{F}_i$ into a stabilized version $\tilde{\mathbf{F}}_i$.

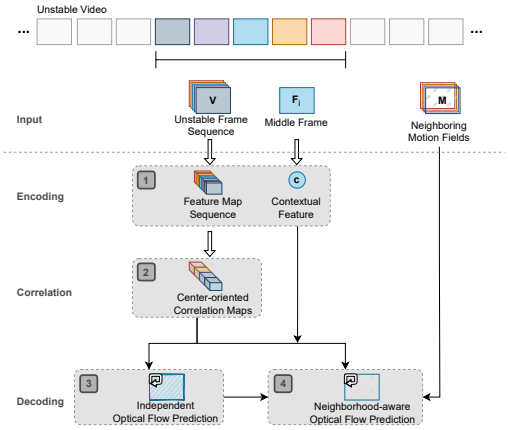


Figure 5. Training Stage. Our training input is a set of unstable frames and a set of neighboring motion fields. The output is the optical flow to stabilize the middle frame. Source: Author's Thesis.

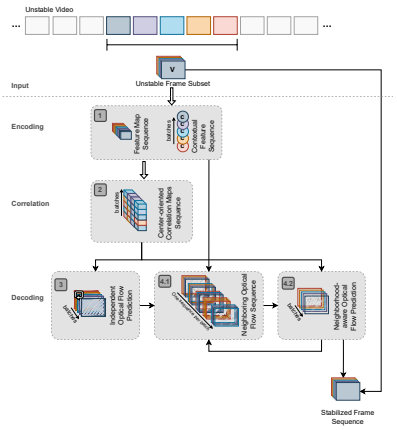


Figure 6. Inference Stage. The input to the inference is a subset of unstable frames. The output is the optical flow (and the stabilized frames) for the subset. Source: Author's Thesis.

The training stage is divided into four steps (Figure 5): (1) generating feature maps for each frame in $\mathbf{V}_i$ and a contextual feature for $\mathbf{F}_i$; (2) calculating $\mathbf{F}_i$-oriented correlation maps; (3) initial iterative decoding of $\mathbf{B}_i$; and (4) final iterative refinement of $\mathbf{B}_i$ incorporating information from neighboring motion fields $\mathbf{M}^{\text{ngb}}$. We employ two terms for supervision during training: a pixel-wise loss between the predicted optical flows and the ground truth motion fields, and a smoothness loss. Notably, both decoders are trained with the same loss function despite their slightly different tasks. This strategy allows the network to implicitly learn full video stabilization from the data, predicting stabilized 3D motion from 2D frame information without requiring explicit assumptions or simplifications.

Figure 6 illustrates the inference stage. We represent the input unstable video as $\mathbf{V} = \{\mathbf{F}_0, \mathbf{F}_1, \cdots, \mathbf{F}_N\}$. Our objective is to compute the sequence of optical flows $\mathbf{B} = \{\mathbf{B}_0, \mathbf{B}_1, \cdots, \mathbf{B}_N\}$. These optical flows are then used to stabilize the frames in $\mathbf{V}$, generating the initial stabilized video estimate $\tilde{\mathbf{V}}^0 = \{\tilde{\mathbf{F}}_0^0, \tilde{\mathbf{F}}_1^0, \cdots, \tilde{\mathbf{F}}_N^0\}$. Optionally, frame boundary masks $\mathcal{M} = \{\mathcal{M}_0, \mathcal{M}_1, \cdots, \mathcal{M}_N\}$ can also be computed. These masks and the warped frames are then fed into a video inpainting method to produce the final stabilized video $\tilde{\mathbf{V}} = \{\tilde{\mathbf{F}}_0, \tilde{\mathbf{F}}_1, \cdots, \tilde{\mathbf{F}}_N\}$.

During inference, unlike the training stage, we compute a contextual map and correlation maps for each frame, arranging them in sequential batches. The second decoder in each iteration uses the neighboring optical flows predicted in the previous iteration instead of relying on the fixed motion fields used during training. These predicted optical flows are then used to warp the unstable frames, generating stabilized versions. Optionally, masks can be computed, and the video is inpainted to refine the final results. Our method operates in a semi-online fashion, processing frames in subsets $\mathbf{V}' = \{\mathbf{F}_{i-\text{anc}}, \cdots \mathbf{F}_i, \cdots, \mathbf{F}_{i+s+\text{la}}\}$ using a sliding window approach. The window size is denoted by s, and the number of anchor frames and lookahead frames is represented by anc and la, respectively. The anchor frames provide context for the current frame, while the lookahead frames allow the model to predict future motion.

4.2. Experimental Results

We assessed the performance of our novel method against five existing techniques. All tests were conducted on NUS Dataset [Liu et al. 2013], containing 144 unstable video clips categorized into six groups based on camera movements and scene characteristics. We classified the existing methods based on their underlying approach. Table 4 summarizes the results, highlighting the best performing methods in terms of frames per second (FPS), model size, and number of learnable parameters.

NAFT achieved FPS comparable to DUT and NAFT-S (smaller version) outperformed it. Additionally, NAFT and NAFT-S exhibited the smallest model size and fewest parameters among all competitors. Compared to existing techniques, NAFT was roughly 20% smaller than the smallest reported DWS method (StabNet) and required approximately 18% fewer parameters. When compared to the overall best performing methods, NAFT's model size was about 63% that of Deep3D, and its parameter number was about 59% that of DIFRINT. Our model size and parameter number were only about 2.3% and 2.1% of those required by StabNet, respectively. Similarly, for the smaller version, the model size and parameter number were approximately 7.5% and 7% of the lowest val-

Table 4. Statistics of Computational Resources.

	Methods	FPS	Size	Params
Others	Deep3D	0.8	36.0	37.2
	DIFRINT	10.6	38.0	9.9
	DUT	4.9	54.4	10.0
DWS	PWStab.	**30.0**	186.0	48.5
	StabNet	13.0	116.0	32.4
	NAFT	4.9	23.0	5.9
	NAFT-S	8.7	**2.7**	**0.7**

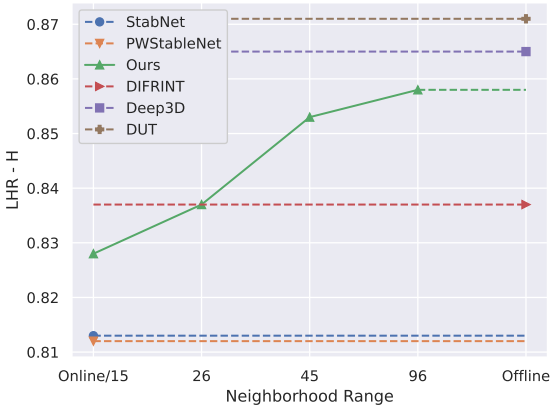


Figure 7. Results with different neighborhood sizes. Markers show the neighborhood used by each experiment. Source: Author's Thesis.

ues reported for Deep3D and DIFRINT, respectively. These results demonstrate that our DWS method achieves comparable performance to non-DWS methods while significantly reducing computational resource consumption in terms of both model size and parameters, especially compared to DWS methods.

We further evaluated the NAFT effectiveness using various neighborhood ranges and compared it with existing methods (Figure 7). Our method consistently outperformed two prior DWS methods (StabNet and PWStableNet) at all neighborhood ranges tested. Notably, NAFT's performance improved as the neighborhood range increased, approaching the stability achieved by the best offline methods (Deep3D and DUT).

An analysis of video stability per category within the dataset revealed that our method consistently outperformed other DWS techniques. The best performing method overall depended on the specific metric used. For instance, based on the LHR-H metric, NAFT achieved the best overall results for videos in the Quick Rotation and Regular categories. According to the LHR-OF metric, NAFT achieved the best results in the Regular and Running categories. In terms of image distortion, our method yielded results similar to those obtained by DUT and Deep3D without using inpainting. However, when inpainting was employed, NAFT achieved the best results in most categories. Additionally, our method produced cropping rates comparable to those achieved by DUT and Deep3D. Detailed results are provided in the full text of the thesis, with a supplementary video available in the following link: https://github.com/marcoosrs/NAFT.

Figure 8 shows a visual comparison of our results with those of literature methods (DIFRINT and FuSta), which revealed that NAFT introduced fewer artifacts while preserving a more realistic appearance.

We identified three main limitations related to video quality: (a) NAFT may introduce spatial distortions in certain frames, particularly in videos with significant instability (Running category); (b) in some cases, NAFT may not correct instabilities as effectively as classical methods; (c) when dealing with large holes or fast video motion, the inpainting process can produce poor results. This is a known limitation of E^2FGVI and other video inpainting methods. Additionally, our method was not the fastest among DWS

(a) DIFRINT **(b)** FuSta **(c)** Ours

Figure 8. Subjective comparison of the sequence of frames filled by FuSta, DIFRINT, and E^{2}FGVI (with fine-tuning). Source: Author's Thesis.

methods, and memory usage was high, potentially causing issues with processing high-resolution videos. Furthermore, our current implementation involves two passes through the network, which is inefficient and can be improved to significantly reduce runtime.

5. Conclusions

This work aimed to improve digital video stabilization by addressing the major gaps in existing research. First, we established a structured framework and taxonomy, and analyzed methods and evaluation metrics, revealing inconsistencies. Second, we proposed a framework for assessing stabilization quality, introducing a method for motion estimation assessment and kinematic measures. We also found recent methods like DWS sometimes performed worse than classical approaches. To address this, NAFT, a stabilization network based on RAFT, was introduced, outperforming other methods while reducing parameters and model size by up to 93%. The training was done on SynthStab, our proposed dataset of over 100K synthetic videos.

References

Grundmann, M., Kwatra, V., and Essa, I. (2011). Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Liu, S., Yuan, L., Tan, P., and Sun, J. (2013). Bundled Camera Paths for Video Stabilization. *ACM Transactions on Graphics*.

Liu, S., Yuan, L., Tan, P., and Sun, J. (2014). Steadyflow: Spatially Smooth Optical Flow for Video Stabilization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Souza, M. R., Maia, H. A., and Pedrini, H. (2022). Survey on Digital Video Stabilization: Concepts, Methods, and Challenges. *ACM Computing Surveys*.

Souza, M. R., Maia, H. A., and Pedrini, H. (2023a). NAFT and SynthStab: A RAFT-based Network and a Synthetic Dataset for Digital Video Stabilization. *Springer International Journal of Computer Vision (under review)*.

Souza, M. R., Maia, H. A., and Pedrini, H. (2023b). Rethinking Two-Dimensional Camera Motion Estimation Assessment for Digital Video Stabilization: A Camera Motion Field-based Metric. *Elsevier Neurocomputing*.

Souza, M. R., Maia, H. A., and Pedrini, H. (2023c). Survey on Digital Video Stabilization: Datasets and Evaluation. *ACM Computing Surveys (under review)*.

Zhao, M. and Ling, Q. (2020). PWStableNet: Learning Pixel-Wise Warping Maps for Video Stabilization. *IEEE Transactions on Image Processing*.