Video Stabilisation Based on Spatial Transformer Networks

Maria Silvia Ito

Department of Electronic Engineering
Queen Mary, University of London

Thesis submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy

June 2021

To my family. . .

Declaration

I, Maria Silvia Ito, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

Maria Silvia Ito
June 2021

Details of collaboration and publications:

Papers published while working on this thesis are listed in Chapter 1. Any publications produced in collaboration with others are clearly mentioned. Any further collaborations are labelled as such within the text.

Abstract

User-Generated Content is normally recorded with mobile phones by non-professionals, which leads to a low viewing experience due to artifacts such as jitter and blur. Other jittery videos are those recorded with mounted cameras or moving platforms. In these scenarios, Digital Video Stabilization (DVS) has been utilized, to create high quality, professional level videos. In the industry and academia, there are a number of traditional and Deep Learning (DL)-based DVS systems, however both approaches have limitations: the former struggles to extract and track features in a number of scenarios, and the latter struggles with camera path smoothing, a hard problem to define in this context. On the other hand, traditional methods have shown good performance in smoothing camera path whereas DL methods are effective in feature extraction, tracking, and motion parameter estimation.

Hence, to the best of our knowledge the available DVS systems struggle to stabilize videos in a wide variety of scenarios, especially with high motion and certain scene content, such as textureless areas, dark scenes, close object, lack of depth, amongst others. Another challenge faced by current DVS implementations is the resulting artifacts that such systems add to the stabilized videos, degrading the viewing experience. These artifacts are mainly distortion, blur, zoom, and ghosting effects.

In this thesis, we utilize the strengths of Deep Learning and traditional methods for video stabilization. Our approach is robust to a wide variety of scene content and camera motion, and avoids adding artifacts to the stabilized video. First, we provide a dataset and evaluation framework for Deep Learning-based DVS. Then, we present our image alignment module, which contains a Spatial Transformer Network (STN). Next, we leverage this module to propose a homography-based video stabilization system. Aiming at avoiding blur and distortion caused by homographies, our next proposal is a translation-based video stabilization method, which contains Exponential Weighted Moving Averages (EWMAs) to smooth the camera path. Finally, instead of using

EWMAs, we study the utilization of filters in our approach. In this case, we compare a number of filters and choose the filters with best performance. Since the quality of experience of a viewer does not only consist of video stability, but also of blur and distortion, we consider it is a good trade off to allow some jitter left on the video while avoiding adding distortion and blur. In all three cases, we show that this approach pays off, since our systems ourperform the state-of-the-art proposals.

Acknowledgements

I would like to thank everyone who took part in this incredible journey. From those who followed closely to those who were more distant. Those who fought with me, and those who fought for me. Those who cheered for me, each one in their own way. To those who supported me, even though they disagreed with some decisions I made. This thesis is for you, for everything you did for me. Without you, I wouldn't be who I am, and I would not have gotten where I am.

First of all, I would like to thank my parents and my family. For everything, always. Then, to my dear friends, both those who stayed in distant lands, and the wonderful friends I made in London. I thank you for your patience, attention, companionship and for the words of encouragement in times of weakness. I also thank my acquaintances. Even if indirectly and discreetly, I learned something from them. This project has the influence of each one of you, some more, others less, but what matters is the contribution to this significant step that I take with this thesis.

Anyway, thank you for everything!

Table of Contents

List of Figures

List of Tables

List of Abbreviations

CDV	Continuous Depth Variation
CNN	Convolutional Neural Network
CTC	Composite Transformation Constraint
DAN	Deep Alignment Network
DDV	Discontinuous Depth Variation
DL	Deep Learning
DLM	Detail Loss Metric
DSO	Direct Sparse Odometry
DVS	Digital Video Stabilization
EWMA	Exponential Weighted Moving Average
F2F	Frame-to-Frame
FPGA	Field-Programmable Gate Array
FPS	Frame Position Smoothing
FR	Full-Reference
FREAK	Fast Retina Keypoint
GAN	Generative Adversarial Network
GT	Ground Truth
HVS	Human VIsual System
IC	Inverse Compositional
IC-STN	Inverse Compositional Spatial Transformer Network
LK	Lucas-Kanade

LST	Locality-Constrained Spatial Transformer
ML	Machine Learning
MLSTN	Multi-Level Feature Fusion Based Locality-Constrained Spatial Transformer Network
MSCOCO	Microsoft Common Objects in COntext
MSE	Mean Square Error
MVI	Motion Vector Integration
NR	Non-Reference
PR	Parameter
RCNN	Recurrent Convolutional Neural Network
ResPrev	Resolution Preservation
ROI	Region Of Interest
RTN	Recurrent Transformer Network
SfM	Structure-from-Motion
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
SSIM	Structural Similarity Index
ST-GAN	Spatial Transformer GAN
STN	Spatial Transformer Networks
SVR	Support Vector Regression
TSTN	Tied Spatial Transformer Network
UAV	Unmanned Aerial Vehicles
VIF	Visual Information Fidelity
VMAF	Video Multimethod Assessment Fusion
VO	Visual Odometry
VSQA	Video Stabilization Quality Assessment

Chapter 1

Introduction

1.1 Motivation

Video stabilization has become a popular topic in the past few years, both in industry and academia. Such popularity is caused by the increase in digital camera utilization, which has led to the recording of unsteady and blurred videos [7]. More specifically, there is a wide variety of videos recorded with smartphones and compact handycams by non-professional users, who tend to induce jitter due to body tremors. There are also a number of videos recorded with mounted cameras for surveillance, law enforcement, and military applications: in this case, the jerkiness is introduced by atmospheric disturbances [8]. Another set of videos are those recorded with moving platforms, such as Unmanned Aerial Vehicles (UAV), robots, and body cameras. In all scenarios, the unwanted camera motion may lead to unpleasant viewing experience by the end user and affect the performance of video processing algorithms, e.g. surveillance and video encoding applications [9]. Therefore, video stabilization is fundamental for generating suitable content for video applications [10].

Video stabilization techniques aim at removing or reducing undesired camera motion from an unsteady video to improve its quality, by generating a compensated video which preserves intentional global motion [11][10][8][12]. Professionals can leverage advanced sensors and lens systems for changing the way the camera receives light or utilize mechanical devices (e.g., tripod, dolly, steadycam) to prevent the camera from shaking during the video capture [13]. In fact, amateur videos differ from professional ones due to two reasons: jitter and blur caused by camera motion [14]. However, the mentioned solutions are unfeasible for amateur utilization, once they are expensive or demand large equipment [8][11]. Other methods rely on gyroscopes for video stabilization, however, gyroscopes can smooth rotational motion, not being able to compensate translation. Added to that, gyroscopes cannot be employed in all levels of cameras as high cost and space requirements prevent them from being used in compact and low cost stabilizers [8][11].

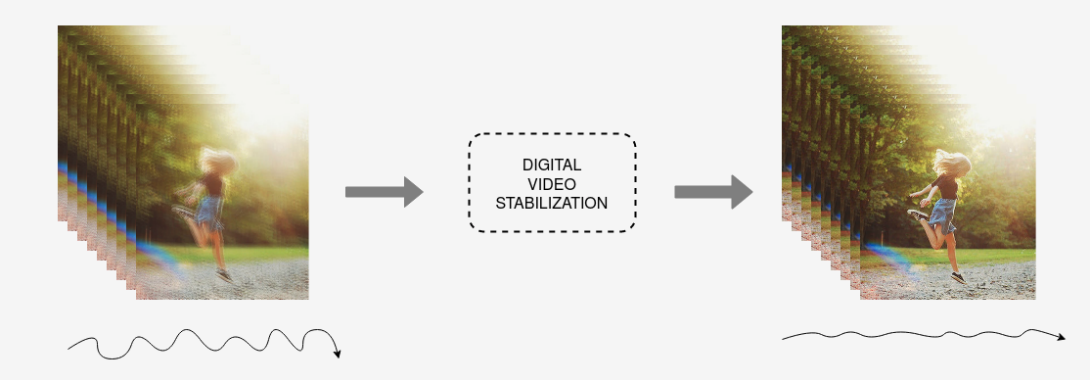


Fig. 1.1 Digital Video Stabilization Overview

Digital Video Stabilization (DVS), seen in Figure 1.1, is more convenient and economical for casual or amateur recordings because it neither requires specific equipment nor demands knowledge of the device used for recording [12][15]. It aims at creating a new video with the same visual content and without unwanted motion components [9]. Several video stabilization systems have been proposed in recent years, to improve the quality of consumer-level videos, which were captured during walking, running,

riding, driving, amongst others [12][15][16]. Post processing for video stabilization normally warps the frames for compensating motion, producing a smooth camera path [17]. Therefore, it provides a better viewing experience and is suitable for a number of computer vision tasks, such as UAV exploration, robot navigation, and others [13].

Traditional DVS methods match pixels, blocks of pixels, or features between neighbouring frames for motion estimation. However, this process is time consuming [11], and for traditional methods, pixels/blocks/features tend to be sensitive to certain characteristics of the video. For instance, quick camera motion and textureless regions tend to produce a low number of matched pixels/blocks/features and a short length of tracked motion [11][9]. In fact, although there has been great advances in the field, traditional methods tend to fail in a number of scenarios, e.g. with occlusion, textureless areas, parallax, dark areas, amongst others [16][15].

The literature has reported that Deep Learning (DL) approaches have demonstrated the ability to address various computer vision challenges, such as super resolution [18], image deblurring [19], style transfer [20], classification[21], recognition[22], amongst others. In this scenario, DL-based approaches for DVS have recently been proposed in the literature [23][24][25][26].

However, video stabilization faces hurdles when it comes to leveraging DL strategies: 1) lack of training data, which would consist of pairs of synchronized steady and unsteady videos with identical route and content. Although these datasets are not required for traditional methods, they are essential for learning-based systems. 2) accurate problem definition, since camera path smoothing is not easily defined in a DL context [23]. 3) shortage of full-reference evaluation frameworks. Next Section, we provide further discussion on the challenges faced in the DVS field, which we will try to address in this thesis.

1.2 Challenges

Digital video stabilization has been a trendy research topic for decades, and there has been significant contributions from industry and academia. However, the available DVS methods still struggle in a number of scenarios, which can be separated into input and output related, as shown in Figure 1.2.

The input related challenges refer to some video characteristics that impair the processing abilities of the algorithms. Textureless objects lead to little amount of features that are found in each frame, which makes it difficult to track them. Parallax, occlusion, and high motion lead to adjacent frames that are too different from each other, which makes it hard for the algorithms to compare consecutive frames, leading to the wrong motion parameters. Also, 3D-based methods tend to struggle with lack of depth in videos, which leads to instabilities in the stabilization process. Lastly, the DVS algorithms tend to face the problem of differentiating the wanted from unwanted motion.

The output related challenges refer to the low quality of the stabilized videos: videos with added blur, distortion, unwanted motion, ghost effect, or videos that require a high amount of zoom, to remove the border effect caused by frame transformation.

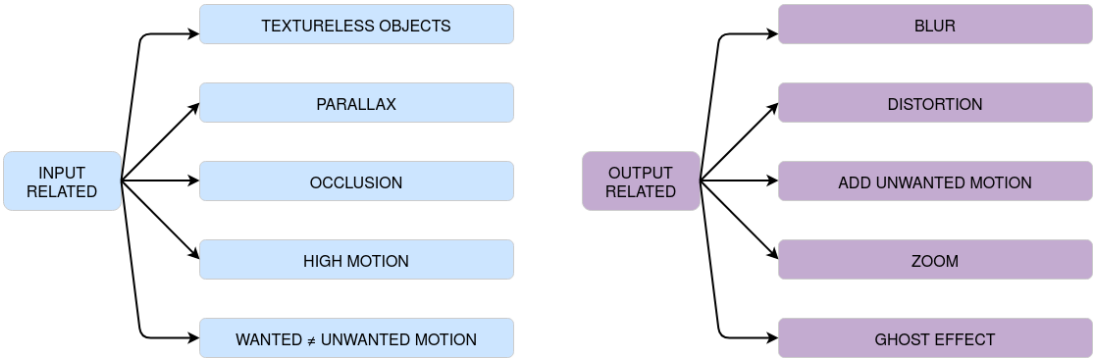


Fig. 1.2 Challenges for traditional DVS methods.

The challenges faced by deep approaches are slightly different (see Figure 1.3). Although they tend to find and track features easily, they still struggle with added blur

and distortion [23][26]. Some approaches [24][[27] have also unsuccessfully tried to generate the frames of the stabilized video. However, the main issue with DL-based DVS methods is the camera path smoothing problem definition, since it is hard to mathematically define a smooth camera path that contains wanted camera motion. With this in mind, we propose our own DVS system, which will be briefly described next Section.

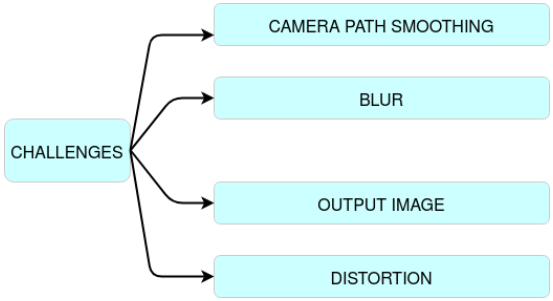


Fig. 1.3 Challenges for deep DVS methods.

1.3 Contributions

This thesis proposes a novel approach for video stabilization, by utilizing the benefits of DL for aligning images and the benefits of traditional methods for smoothing the camera path. Our approach addresses the challenges presented in Figures 1.2 and 1.3.

1) Having in mind the dataset shortage and the fact that current DL-based DVS systems estimate transformation matrices in their algorithms, we provide a dataset that contains such information in Chapter 3. We leverage the knowledge provided by the dataset in [23] to produce a large dataset of artificial steady-unsteady video pairs. Unlike [23], we provide motion parameters, which can decrease the complexity in the training process of future algorithms, since the loss function can be based on the motion parameters instead of the output frames.

2) Performance evaluation in previous studies have mostly been non-reference (i.e., taking only the stabilized video into account). Since we provide a large amount of Ground Truth (GT) and parameter (PR) data, we implement a full-reference evaluation framework in Chapter 3, for performance evaluation based on GT, PR, and stabilized data. To demonstrate the practical use of the introduced dataset and the evaluation framework, we compare the performances of Estadeo [28], a 2D-based DVS algorithm, and StabNet [23], a DL-based DVS algorithm.

3) We leverage Spatial Transformer Networks (STNs) [29] to learn motion parameters between frames in Chapter 4. We simplify the problem by, initially, learning the transformation parameters between two versions of a given image, one of them being previously warped. After obtaining accurate results, we move on to the next step, by using this model to stabilize videos. Our training procedure considers images, therefore our proposed system does not require video datasets that are specific for training DVS systems, which simplifies the process. For testing, our system requires single unstable videos. Because we use the evaluation framework we have provided in 2), in the performance evaluation we also utilize our dataset[1], mentioned in 1).

4) In our first DVS approach, we use moving averages and full homographies to stabilize videos in Chapter 5. To this end, we perform motion estimations between frames and provide two outputs, the first one being long-term smoothed and the second one being long and short-term smoothed.

5) Aiming at avoiding blur and possible distortions caused by homographies, we propose a translation-based DVS system in Chapter 6. This approach uses Exponential Weighted Moving Averages (EWMAs) and moving averages.

[1] https://github.com/mariito/DVS_

6) Since in the literature there are a number of filters utilized for smoothing the camera path, in Chapter 7 we propose a DVS approach that uses filters. We present a thorough analysis and performance comparisons between a number of low-pass filters and their parameters, to select one, or a few, that present best overall performance. Such filter selection is performed considering the homography-based and translation-based approaches.

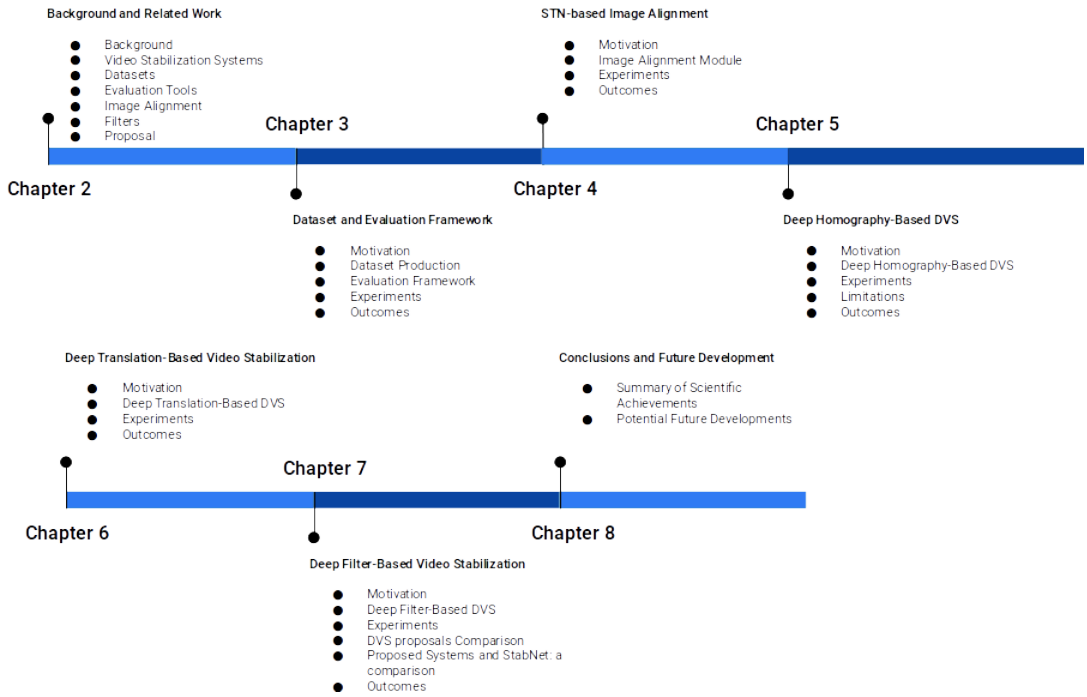


Fig. 1.4 Thesis structure.

1.4 Thesis Structure

The remainder of the report is organized as follows and is shown in Figure 1.4. Chapter 2 presents a brief background and related work on video stabilization and deep learning. Chapter 3 introduces the dataset production and our evaluation framework. In the same Chapter, we describe other datasets and metric that we use in this thesis. Chapter 4 presents our image alignment module, based on STNs. Chapter 5 presents our deep homography-based DVS system. Chapter 6 presents our deep translation-based DVS

systems. Chapter 7 presents our deep filter-based DVS. Finally, Chapter 8 presents our concluding remarks and potential future developments.

1.5 Published Work

- "A Dataset and Evaluation Framework for Deep Learning Based Video Stabilization Systems," 2019 IEEE Visual Communications and Image Processing (VCIP), Sydney, Australia, 2019. In this thesis, the given work is presented in Chapter 3.

- "Distortion-Free Video Stabilization," 2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), London, United Kingdom, 2020. In this thesis, the given work is presented in Chapters 4 and 6.

- "Deep Homography-Based Video Stabilization," 2020 International Conference on Pattern Recognition (ICPR), Milan, Italy. In this thesis, the given work is presented in Chapters 4 and 5.

Chapter 2

Background and Related Work

This Chapter presents a background (Section 2.1) followed by related work on video stabilization systems (Section 2.2), datasets (Section 2.3), evaluation tools (Section 2.4), image alignment (Section 2.5), low-pass filters (Section 2.5.4), and finally our project proposal (Section 2.7).

2.1 Background

This Section presents a Background on Digital Video Stabilization and Deep Learning.

2.1.1 Video Stabilization

DVS has been a research topic for decades, presenting a wide variety of techniques and characteristics. Because of this, DVS systems can be classified according to:

- The time the videos are stabilized [13]

    - **Offline:** after the videos are recorded

- **Deferred:** while they are being recorded, with a delay between input and output frames

- **Real-time:** while they are being recorded, without negligible delays

Because motion estimation algorithms are computationally expensive, most DVS systems are offline or deferred. Also, there are a number of DVS methods that utilize both past and future frames of a recorded video for high quality stabilization. However, the need for real time stabilization has raised recently, for applications that demand interaction between the video application and the viewer. For example, robots and drones that communicate with a remote operator, who can ideally view a stable video in real time to make immediate decisions [30]. Therefore, real time video stabilization is a challenge [13].

- The complexity of the model used to process the frames [15][31]:

  - **2D-based**: utilizes 2D transformations, such as affine and homography, to model and smooth the camera path. These methods are fast, robust, and demand low computational cost. However, they are limited at handling parallax, relative depth variations, large or abrupt camera movements, camera translation, and scaling [32][9].

  - **3D-based**: utilized for reconstructing and smoothing 3D camera trajectories, allowing representation of realistic displacements in all directions. The motion estimation step strongly relies on depth estimation, which is slow and computationally expensive. Although they can deal with complex geometry, they do not deal with lack of depth, parallax and occlusion well, as they tend to introduce ghost effects on the stabilized frames [31][9].

- **2.5D-based**: leverages partial 3D information for video smoothing by relaxing the 3D reconstruction. They aim at obtaining the correctness of 3D based techniques while more computationally efficient than 3D approaches [15][11].

- **Hybrid**: leverages the strengths of 2D and 3D methods, by applying 2D methods to certain frames and 3D methods to other frames [15].

- **Deep**: utilizes Deep Learning strategies for video stabilization [23][24].

Selecting the appropriate model can be tricky, since the mentioned models perform differently, depending on the scene and camera path. Choosing an inappropriate model can cause low quality video stabilization results [9].

Generally, DVS methods are performed in three steps[10][17][11], as shown in Figure 2.1:

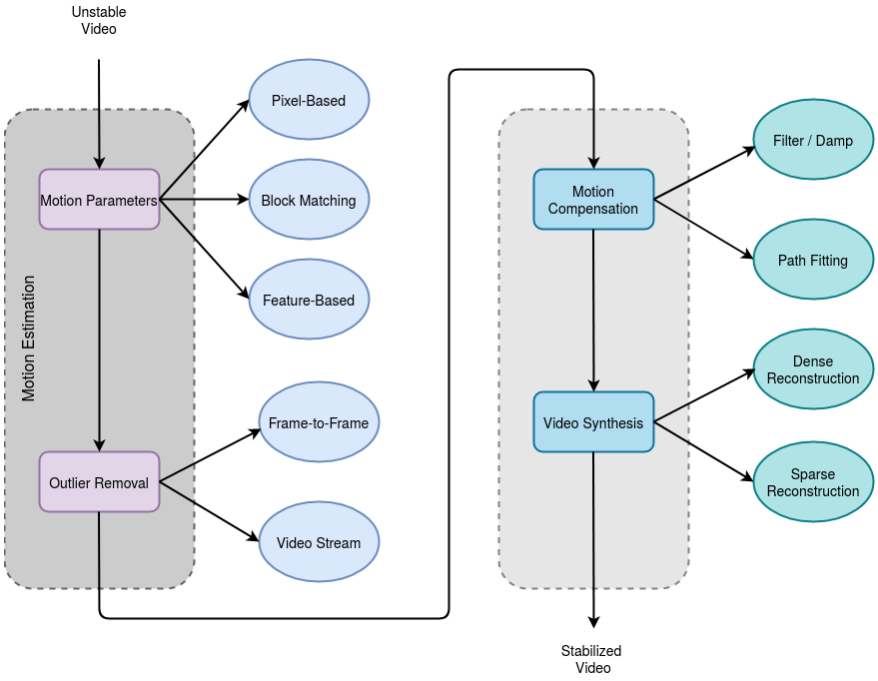


Fig. 2.1 DVS methods steps.

1) motion estimation: fundamental for stabilization methods, it analyzes the video to estimate camera motion, since it differentiates camera motion from moving subjects/objects. Initially, the camera **motion parameters** are estimated with the spatio-temporal correspondence between adjacent frames. It can be done by matching pixels (or blocks of pixels, or even features) between consecutive frames with either a similarity or distance metric [9]. Then, the camera motion is differentiated from the movement of subjects or objects with **outlier detection and removal**. It can be done either on a frame-to-frame basis or by analyzing the video stream. After outlier removal, the remaining motion can be used to model or approximate the camera motion per se.

However, pixel/block/feature matching and tracking are time consuming. They may also be sensitive to camera motion (e.g., quick camera motion) and scene content (e.g., textureless regions, occlusion) [10][11][16]. These scenarios may lead to a limited amount of matches or to a short duration of tracked motion.

2) motion compensation: aims at removing or reducing the unwanted camera motion and to compute a new plausible camera motion. The intensity of the stabilization algorithm can be adapted, so as to provide a good viewing experience [9]. Since one of the most common causes for discomfort are high and/or low frequency jitter, one solution is to use filters or damping to remove such motion. These filters can be applied to pixels/blocks/features trajectories or to camera motion parameters. Particularly, there is a difference in the use of filters or of damping techniques:

i) Frame Position Smoothing (FPS) uses low-pass filters to smooth the absolute frame position that is obtained with accumulated global motion vectors. These are constructed by accumulating global inter-frame motion vectors from an unstable video. The accumulated motion vectors are low-pass filtered, to remove high frequency components caused by jitter and maintain low frequency components, caused by intentional camera

movement. The low-pass filtered motion vector $X_{lpf}(n)$ and the original accumulated global motion vector $X_{act}(n)$ are, then, used to estimate the corrected vector $V_{cor}(n)$ [33].

$$V_{cor}(n) = X_{lpf}(n) - X_{act}(n) \tag{2.1}$$

ii) Motion Vector Integration (MVI), generates correction vectors by integrating the differential global motion vector of a frame using a damping coefficient k [33].

$$V_{\text{int}}(n) = k \times V_{\text{int}}(n-1) + V_{act}(n) \tag{2.2}$$

Even though the community has widely utilized MVI, [33] has proven that FPS is better than MVI.

Another option is to use path fitting, which consists of fitting the camera motion to a particular model: constant models replicate still shots, linear models replicate tracking shots, and quadratic models can replicate a transition from one to another. However, this approach is not applicable to long video sequences and may require user input to select the desired motion type [9].

3) video synthesis: generates a new video that corresponds to the smooth camera path, by applying the newly smoothed camera motion to the original unstable video. This step depends on the camera model of choice [9]. While some models describe motions for all pixels in the frames, using dense reconstruction, some describe motion for specific points, with sparse reconstruction.

These blocks are not necessarily present in all proposed methods, and may present some differences to the described steps. However, the mentioned steps constitute a convenient and straightforward way to describe the video stabilization pipeline.

2.1.2 Machine Learning and Deep Learning

Machine Learning (ML) algorithms aim at representing the input data and generalizing the learned patterns for utilization on unseen data. With ML, computers can act without explicitly being programmed, with algorithms that learn from data, by making data driven decisions and predictions [34]. The quality and fit of the data highly influence the performance of the ML algorithm. Therefore, feature extraction and data representation from raw data are crucial to ML. In fact, these procedures take up a large percentage of the effort in a given ML task. Also, they are domain specific and require substantial human intervention [35]. Although traditional ML has been widely used in pattern recognition, natural language processing, and computational learning, when it comes to speech and vision, its performance is far from satisfactory. DL was introduced in the past years to fill this gap [34].

Deep Learning (DL), a sub-field of ML, is an emerging approach which aims at learning high level abstractions in data by leveraging hierarchical architectures [36][37]. The goal of DL is to automate the extraction of complex data representations. These algorithms consist of a layered and hierarchical architecture for learning and representing data. In this hierarchy, higher level, or more abstract features are defined based on lower level, or less abstract features [36][35]. The main idea is that these layers are not designed by engineers: instead, they are learned from the input data with a general purpose learning algorithm [38]. Such hierarchical architecture is an attempt to emulate the learning process of the human brain [35].

DL methods have provided good results in different ML fields, such as object detection and recognition [39], media retrieval, multimodal data analysis, speech recognition, computer vision, natural language processing, image analysis, video tracking, amongst others [36][34][35].

One of the most notable DL approaches is the Convolutional Neural Network (CNN), being the most commonly used in several computer vision applications [37]. CNNs generally consist of three main layers [37]]: i) convolutional layers: in these layers, several kernels or filters convolve the whole image and the intermediate feature maps, creating several feature maps, as shown in Figures 2.2 and 2.3. This type of layer allows weight sharing within the same feature map, leading to a drop in the number of parameters ii) pooling layers: they usually follow a convolutional layer, and are utilized for reducing the dimensions of the feature maps and the network parameters. The most common approaches are max pooling and average pooling iii) fully connected layers: responsible for converting 2D feature maps into a 1D feature vector, performing like traditional neural networks. They allow feeding a neural network into a vector with pre-defined dimensions. However, they contain many parameters, leading to high computational cost for training

In 2D CNNs (shown in Figure 2.2), convolutions are applied on 2D feature maps, only computing features from spatial dimensions. On the other hand, 3D CNNs (shown in Figure 2.3) are used to compute features from spatial and temporal dimensions, e.g., in a video. In this case, the convolutions are applied convolving 3D kernels to cubes formed by stacking multiple consecutive frames together. With this structure, the feature maps in a given convolution layer are linked to several consecutive frames in the previous layer, allowing the extraction of motion information. 3D CNNs are widely used in video analysis due to this reason [40].

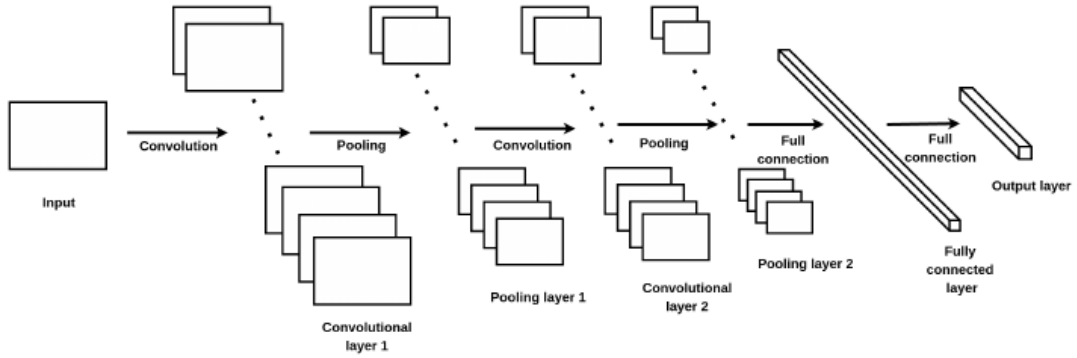


Fig. 2.2 2D Convolutional Neural Network Structure. Extracted from [1]

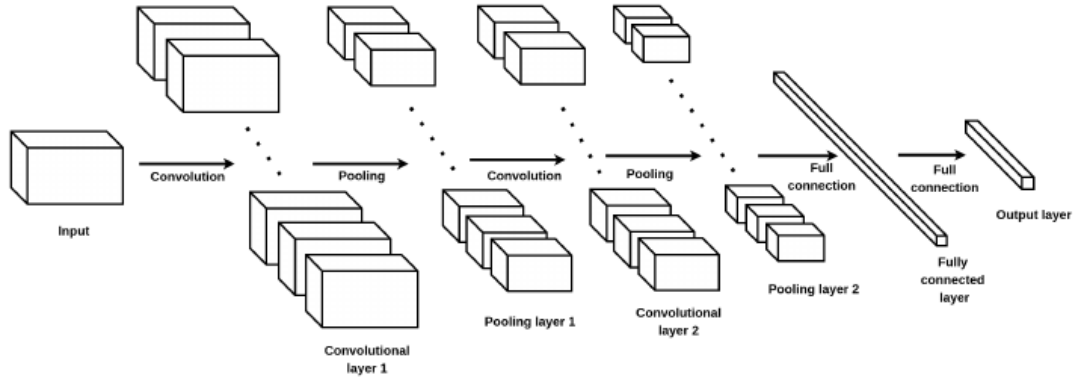


Fig. 2.3 3D Convolutional Neural Network Structure. Extracted from [1]

Although CNNs present good performance in a variety of computer vision tasks, they are not spatially invariant to the input data in a computationally and parameter efficient way [24][29]. For spatial manipulation of data within the network, Spatial Transformer Networks (STNs) have recently been proposed [29]. Spatial Transformers can be utilized in standard neural network architectures to provide spatial transformation capabilities. The spatial transformer does not require extra supervision, and can spatially transform an image, or feature map, by providing transformation parameters for each input. The transformation provided is performed on the entire image, or feature map, and can include scale, crop, rotation, and non-rigid deformation. Spatial Transformers can be inserted into CNNs for performing a number of tasks, namely image classification, co-localization, spatial attention, amongst others [29]. Another

task is warp learning, in which it learns a set of warp parameters, followed by a differentiable warp function [41].

Spatial transformers are differentiable modules that apply a spatial transformation to a feature map, producing a single feature map as output. In case of an input with multiple channels, the same transformation is applied to all channels. As shown in Figure 2.4, a spatial transformer consists of three parts: 1) localization network, which takes the input feature map and, through a group of hidden layers, outputs the transformation parameters that should be applied to the feature map. The predicted parameters are, then, passed to the 2) grid generator, which uses the parameters to create a sampling grid, i.e., a grid of points where the input map should be sampled to generate the transformed output. Lastly, the input feature map and the sampling grid are received by the 3) sampler, which produces the output feature map [29].

STNs have been utilized in a number of tasks: facial recognition [42], inverse composition [43], neural circuits reconstruction [44], deformable image registration [45], and video stabilization [24][46], amongst others.

In the past years, Deep Learning techniques have provided good results in different computer vision fields, such as super resolution [18], image deblurring [19], style transfer [20], medical imaging [47], among others. However, Deep Learning based solutions for video stabilization have not been thoroughly studied yet. This is mostly due to: 1) lack of training data, which would consist of pairs of synchronized steady and unsteady videos with identical route and content. Although such datasets are not required for traditional methods, they are normally required for learning-based systems. 2) accurate problem definition, since camera path smoothing is not easily defined in a CNN context [23].

2.2 Video Stabilization Systems

Digital video stabilization has been a research topic for decades, to the best of our knowledge dating back to 1998 [48], as the use of amateur digital handheld cameras started to spread. Along the years, digital cameras have evolved [1] and their utilization has become more frequent, especially after the start of the smartphone utilization [2]. Not to mention the increase in security cameras[3] and devices equipped with cameras (such as drones and robots). With the available amateur recording devices, video stabilization has also evolved, starting with 2D approaches, followed by 3D, 2.5D, and hybrid approaches. Next, a few DL-based approaches have been proposed. In this Section, we present papers related to some of these approaches.

Although there are a number of different classifications for DVS systems, in this thesis, we focus on the most utilized one, which focuses on the model complexity, separating different proposals as 2D, 2.5D, 3D, hybrid, and deep.

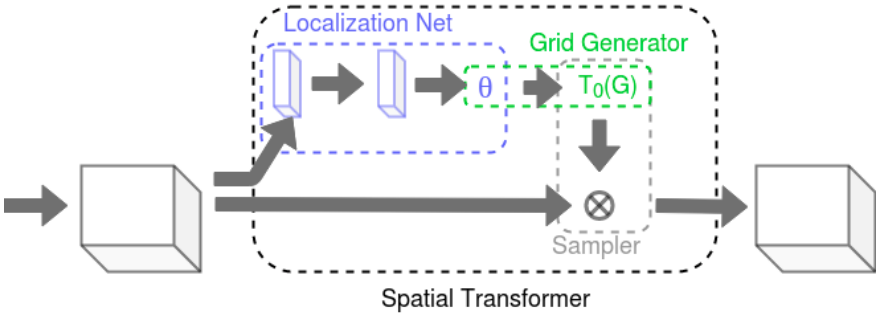


Fig. 2.4 Spatial Transformer Network.

2.2.1 2D-based

2D-based DVS methods estimate 2D transformation (e.g., affine and homography) parameters between adjacent frames and concatenate such parameters to form a 2D

[1] http://uk.businessinsider.com/digital-photography-revolution-2015-4?r=US&IR=T
[2] https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/
[3] https://www.statista.com/statistics/477917/video-surveillance-equipment-market-worldwide/

camera path [49][50][51] or a grid of 2D camera paths [50][30]. The optimal path consists of temporally smoothed transformations [23]. It is estimated with low-pass Gaussian filter [52][53][54], Kalman filter [55][56], least square fitting [57], low-pass filters [58][59], L1-norm optimization [49][60], bundled camera paths [50], geodesics optimization [17], or rolling shutter effects analysis [61]. Early 2D methods estimated homographies between adjacent frames and then smoothed all obtained homographies after concatenating them to obtain stable frames. Since 2D methods match features between adjacent frames, they are more robust to camera motion and efficient than 3D models [23][11].

2D methods are known to be easily implemented, but prone to artifacts (e.g., wobbling, distortion, blur), especially for videos with parallax (caused by depth variation or out of plane motion [17]). In these cases, the distortion added is likely to be more annoying to the viewer than the jitter from the original video. Also, they are still sensitive to camera motion (e.g. quick camera motion) and scene content (e.g. textureless regions and occlusion) [10][11][16].

State-of-the-art 2D methods either estimate the unsteady camera path with global [52]][49][55] or local [50][62] parametric 2D transformations between consecutive frames, then optimize the estimated parameters to obtain a smooth camera path.

Grundman et. al. [49] adopt some smoothness constraints based on cinematographic criteria, and utilize L1-norm optimization for affine transformations to obtain a smooth path, enabling high quality video stabilization results. The authors prove that a well defined path can lead to good stabilization results even with 2D transformations. However, since the optimization does not preserve shape, it can still lead to distortion.

Liu et. al. [50] propose a 2D stabilization method that considers bundled camera paths. In this model, each frame is divided into a mesh grid, and each cell has its own path,

allowing it to handle non linear motion caused by parallax and rolling shutter effects. Also, the model requires feature matching between two consecutive frames, allowing the robustness and simplicity of 2D methods. However, the model has its drawbacks. According to the authors, the warping can fail in the presence of severe occlusion, especially if combined to rolling shutter effect. In this case, it prefers to leave some jitter to avoid distortion. Also, the path optimization does not obtain cinematographic results, which can be undesirable in some applications.

Estadeo [28] is a paper dedicated to implementing and exhaustively comparing classic DVS techniques and boundary conditions. It also analyzes two algorithms that aim at cropping the videos after they are stabilized. Since the author has provided the source code to the system, in this thesis we extensively use it as baseline. To this end, we utilize the default options in the code provided by the author.

Battiato et al. [63] present a video stabilization algorithm based on the extraction and tracking of Scale-Invariant Feature Transform (SIFT) features through video frames. The proposed approach contains a modified version of iterative least squares method to avoid estimation errors and intentional camera motion is filtered with Adaptive Motion Vector Integration.

Kim et al. [64] propose a method to estimate the global motion accurately using the background and foreground feature points. In the approach, global motion estimation and feature point classification are performed simultaneously using the feature point correspondences and the global motion parameters of the previous frame.

Su et al. [65] propose an approach which combines the techniques of video stitching and stabilization into an optimization framework. The considered scenario consists of two videos captured with mobile devices recording the same scene, from different

angles, which are then stitched together. The proposed method computes the optimal stabilization and stitching results with respect to each other.

Hu et al. [66] present a method that contains a multilayer gray projection algorithm to estimate translation, rotation and scaling motion between target and reference images. Differential gray projection is applied to estimate relative image translation, then the scaling and rotation are calculated by doing ring projection and circular projection.

Chen et al. [67] propose an approach that is based on the shortest spanning path clustering algorithm for estimation of the global motion vectors. Feature points detected by Fast Retina Keypoint (FREAK) descriptors [68] from each frame pair are used to obtain global motion vectors through the shortest spanning path clustering algorithm. Then, these vectors are used to stabilize the successive frames.

Dong et al. [69] propose an instantaneous video stabilization method for Unmanned Aerial Vehicles (UAV). The proposed method smooths the video motion in both two-dimensional and three-dimensional scenes and instantaneously provides the stabilized footage to users. For each input frame, the approach applies inter-frame transformations to the four corners of the image rectangle, then uses an adaptive filter to smooth motion trajectories.

Kejriwal and Singh [70] also propose a video stabilization algorithm for UAV. The proposed system extracts corner points with Good Features to Track corner detection algorithm. Then, optical flow between consecutive frames is computed with the extracted points. The optical flow detected points are used to estimate motion parameters using an affine transform model. Finally, a hybrid filter consisting of Kalman and low-pass filter is used to smooth the estimated motion parameters and the frames are warped using the smoothed parameters to obtain a stabilized video sequence.

Walha et al. [71] present a video stabilization and moving object detection system. It uses local feature extraction and matching to estimate global motion with SIFT keypoints. The proposed approach estimates global camera motion using affine transformation, then uses Kalman filter to detect moving objects. To smooth the camera path, it uses a median filter to retain the desired motion.

Although the presented methods tend to produce good results, few of them can provide real-time stabilization for portable devices due to their high computational cost. A number of Field-Programmable Gate Array (FPGA) implementations have been proposed for video stabilization to date [72][73][74]. However, they can only address videos with translational jitter, not being able to address rotation and scaling. The work in [10] is another FPGA implementation, feature-based full-frame stabilization method. It consists of 2D method which estimates affine transformations between consecutive frames: it can deal with translation, rotation, and scaling.

2.2.2 3D-based

3D-based methods estimate the camera path based on the full reconstruction of the camera path in the 3D space, by utilizing Structure-from-Motion (SfM) [75][76][77] or depth sensors for 3D reconstruction [78], to obtain rotation and translation information to model camera motion. The 3D camera path is, then, smoothed and the unsteady frames are rendered with the new smoothed camera configuration, generating stabilized frames. These methods provide good stabilization results, however they are usually more computationally expensive and brittle than 2D methods [17][15]. Finally, 3D methods might fail due to tracking errors (since they normally utilize long feature tracks for 3D structure reconstruction, including scene points and camera motion [15]) and scene degeneration[11].

Liu & Jin [75] propose a 3D method of video stabilization that simulates an ideal camera motion from a handheld recorded video. The model is based on existing 3D proposals, but can avoid ghosting effects by warping each output frame from a single input frame. For guidance on the warping process of the output, they utilize 3D sparse point cloud. The proposed system presents higher resulting stability than traditional 2D models while avoiding the ghosting effect found in previous 3D proposals. Overall, the method is still more time consuming than 2D models due to the 3D point clouds, and can suffer in the absence of camera translation. Also, the output frame needs to be cropped, and due to the aggressive warping the content loss due to croppping might be higher than other techniques.

Lee et al. [79] propose a fast video stabilization algorithm that provides reduced computational complexity. The proposed approach estimates the 3D information of the feature points in each input frame and defines the Region Of Interest (ROI) based on the obtained 3D information. Then, if the number of feature points in the ROI is sufficient, it applies ROI-based pre-warping and content-preserving warping sequentially to the input frame. Otherwise, conventional full-frame warping is applied.

We noticed there is a disagreement in the literature with respect to subspace constraints [80] and epipolar geometry [32]. In this work, we consider these as 2.5D methods.

2.2.3 2.5D-based

2.5D-based methods leverage partial 3D reconstruction of the camera path by relaxing some requirements adopted in 3D reconstruction [11][80]. Some constraints utilized are epipolar geometry [32], mixed homographies [81], local similarity transformations [16] or homographies [50][62], space-time constraints based on free-form curve fitting [82], steady optical flow [51] or subspace constraints [80]. Although these methods tend to produce good results with low computational cost, they are prone to failure

under severe jitter [17][15] and in the presence of occlusion or textureless areas [16]. Also, since the optical flow methods require reliable optical flow estimation, they are prone to failure in the presence of occlusion or textureless areas [16].

Liu et al. [83] present a joint subspace stabilization method for stereoscopic video. The authors initially prove that the low-rank subspace constraint for monocular video proposed by [84] holds for stereoscopic video, since the feature trajectories from the left and right video share the same subspace. Then, they propose a DVS method that computes a common subspace from the left and right videos and uses it to stabilize the two videos simultaneously. The proposed method does not require 3D reconstruction or explicit left-right correspondence to meet the stereoscopic constraints.

Liu et al. [80] transform a set of input 2D camera paths so that they are both smooth and visually plausible. To this end, he authors achieve this by enforcing subspace constraints on feature trajectories while smoothing them.

2.2.4 Hybrid-based

Hybrid methods leverage the strengths of 2D and 3D methods, by applying 2D methods to certain frames and 3D methods to other frames [15]. Because they leverage 2D and 3D methods simultaneously, they present the same drawbacks as these methods. Hybrid architectures tend to achieve better robustness, speed, and smoothing ability than either pure 2D or 3D methods can provide.

Liu et. al. [15] employ a hybrid video stabilization approach for videos that is robust to various camera motions and can handle Discontinuous Depth Variation (DDV), a challenging scenario in video stabilization. According to the authors, 3D methods can handle DDV well, with the computational burden and the long processing time as drawbacks. They automatically partition the video frames into DDV and Continuous

Depth Variation (CDV), and employ 3D and 2D stabilization methods to the frames with DDV and CDV, respectively. Then, the frames are grouped with an optimization constraint.

Johannes [85] presents a hybrid 3D-2D algorithm for stabilizing 360° video using a deformable rotation motion model. The approach estimates the rotation between key frames that are appropriately spaced using 3D analysis. It uses 2D optimization to maximize the visual smoothness of feature point trajectories for the remaining frames. The algorithm contains a rotation motion model that enables handling small translational jitter, parallax, lens deformation, and rolling shutter effect.

2.2.5 Deep Learning-based

The literature has recently reported that Deep Learning (DL) approaches have the ability to address various computer vision challenges, such as image recognition [86][87] and image generation [88][89], optical flow [90], camera motion estimation [91], action recognition [92], scene dynamic generation [93], frame interpolation [94], deblurring [19][95]. In this scenario, DL-based approaches for DVS have recently been proposed in the literature [23][24][25][26][27].

StabNet [23] is a deep online video stabilization method which considers a few past stabilized frames for stabilizing future frames. Instead of explicitly estimating and smoothing camera paths, it leverages a CNN to predict warp parameters. This is performed by learning multi-grid transformation parameters for each unstable frame, using only past frames. StabNet can stabilize low quality videos, with night scenes, watermarks, blurry and noisy videos, which tend to lead existing methods to failure. However, StabNet is prone to failure in the presence of extreme near-range videos or drastic motion.

The work proposed in [24] presents a novel deep online learning framework that learns transformation parameters for each unsteady frame, considering historical stabilized frames. It consists of a generative network with embedded STNs in different layers of the network. The work also contains an adversarial network, which determines the stability of a video fragment. The system can stabilize low-quality videos, scenarios in which traditional methods tend to fail (i.e., heavy noise or multiple exposure). However, the generated affine transformation only considers transformation from the previous frame, resulting in a weak temporal coherence.

Yu et al. [27] use optical flow between adjacent frames to generate dense correspondence of all pixels between the two frames. Then, the video is stabilized by minimizing the distances between corresponding pixels. However, this is a large scale non-convex problem, which is hard to solve. The authors' approach is to use CNNs purely as an optimizer rather than learning from data: the proposed method trains the CNN for every unstable video, and intentionally overfits the CNN parameters to produce the best result. It is robust to scenarios that would lead state-of-the-art solutions to failure. The authors argue that this is a viable formulation for video stabilization, however it is limited by the need to train and overfit a CNN for every input video, leading to a high computational time.

Choi et al. [26] propose an unsupervised framework that utilizes frame interpolation to stabilize frames, which eliminates cropping. The proposed network generates a frame that would be between two sequential unstable frames. Since the middle frame would be between two unstable frames spatially and temporally, generating a sequence of middle frames would result in a more stable video. However, since the proposed work performs a number of interpolations for stabilizing videos, such an approach may generate blurred videos.

Zhao et al. [96] propose a cascade of encoder-decoder architectures that receives a number of unstable frames for generating precise warping maps, with a high degree of freedom. The authors also propose an effective method to automatically determine the cropping region. However, since the proposed method is trained with the DeepStab dataset [23], it has mainly learned how to handle the jitter present in the dataset, and may present degraded performance when handling videos with different movement patterns. Also, the cropping algorithm may inaccurately determine the cropping region.

There have been proposals that leverage a siamese CNN [23], a Generative Adversarial Network (GAN) with embedded STNs [24], weight optimization [25], and frame interpolation [26]. However, as pointed by [6], DL-based video stabilization methods still have room for improving their performance. Also, the stabilized frames in [23][24][25] are a result of 2D transformations, which adds blur and distortion to the resulting video, and [26] performs cascaded frame interpolations to stabilize videos, which can add blur and noise.

Table 2.1 shows a summary of the current DL-based DVS systems. As one can see, their output videos present added artifacts, normally blur and distortion, caused by the use of 2D transformations or interpolation. It is clear that DL-based DVS systems have not yet managed to seamlessly produce stable videos, since the logistics behind the camera path smoothing is still complex and does not produce adequate results. Hence, we argue that there is room for improvement in DL-based DVS.

2.3 Datasets

Although video stabilization is a research problem being investigated for decades, more promising Deep Learning based solutions have not been thoroughly studied yet, mainly because of the shortage of datasets for training, camera smoothing problem

Work	Network	Main feature	Advantage	Disadvantage
StabNet [23]	Siamese CNN	Learns multi-grid transformations	Can stabilize low quality videos, with night scenes,watermarks, blurry and noisy videos	_Prone to failure with extreme near-range videos or drastic motion _Performs 2D transformation, adding blur and distortions
Xu et al. [24]	GAN with embedded STNs	Contains a generative network with embedded STNs in different layers of the network	Can stabilize low-quality videos	_Weak temporal coherence _Performs 2D transformation, adding blur and distortions
Yu et al. [27]	CNN	Uses optical flow and the CNN as an optimizer	Uses optical flow estimation to generate a warp field for each frame	_Requires training and overfitting the CNN for every input video, leading to a high computational time _Performs 2D transformation, adding blur and distortions
Choi et al. [26]	CNN	Uses frame interpolation	Avoids cropping the frames in the stabilized video	_The high number of interpolations may add blur and distortions to the output videos
Zhao et al. [96]	CNN cascade	Uses a cascade of encoder-decoder networks for estimating warping maps	The warping maps present a high degree of freedom Effective cropping algorithm	_Has mainly learned how to handle the jitter present in the DeepStab dataset [23] _The warped maps require interpolation, which adds blur and distortions to the output videos

Table 2.1 Deep Learning-based DVS systems in the literature.

definition [23], and shortage of evaluation frameworks [6]. In fact, most DVS systems proposed until 2018 were either tested on synthetic data or real data without ground truths. According to [9], the first significant dataset composed of stable/unstable video pairs was published in 2018 [97]. Since then, other large scale datasets have been made available [23][6], including the one we present in Chapter 3.

There are a number of DVS systems that provide unstable videos for testing [75][49][80] [98][50][99][100][97][23]. However, out of these only [97] and [23] provide ground truth videos. Both have recorded synchronized video pairs with a handheld device, the former providing 45 videos (categories: walking, climbing, running, riding, driving, large parallax, crowd, near-range object, dark) and the latter providing 60 videos (cate-

gories: Simple, blurry, high motion, dark, textureless, parallax, discontinuous depth, crowd, close object).

However, the amount of videos provided by both proposals is not enough for training and testing DL-based approaches. In fact, for training StabNet (proposed in [23]), data augmentation techniques were utilized to the dataset. Also, the dataset lacks the corresponding motion parameters between the stable-unstable video pairs. The dataset presented by [97] presents the same issue: not enough videos, which was solved by adding synthetic videos to the Full-Reference (FR) tests performed by the authors in the same work. An alternative to recording an entire dataset is to synthetically produce unsteady videos, which has been done in [101][102]. However, these works have not provided a dataset.

2.4 Evaluation Tools

Video Stabilization Quality Assessment (VSQA) evaluates the performance of DVS systems in terms of perceptual quality. It is considered a multi-criterion problem, since factors such as camera motion and artifacts caused by the DVS system (e.g., resolution loss, distortion, blur) contribute to the final quality of experience. These factors are not easily described mathematically [9]. Although a number of DVS systems have been proposed for decades, there is currently no well defined and accepted framework for performance evaluation of DVS systems.

In the literature, there are two approaches that have been used to assess the quality of DVS systems: 1) with visual inspection or user studies. However, these subjective tests are normally time consuming and difficult to setup, being therefore unfeasible. 2) with objective metrics. In this scenario, one can carry out VSQA in two ways, depending on the availability of a ground truth video: i) full-reference: analyzes the quality of a

DVS system by comparing the stabilized and the ground truth videos. ii) non-reference: leverages statistical models for assessing the quality of a DVS system.

However, both approaches pose challenges: 1) it is difficult to obtain ground truth videos for pairwise comparisons. 2) motion stability is a perceptual concept, therefore it is difficult to be modelled mathematically [12].

In previous state-of-the-art DVS systems proposals, [23][24][50], the utilized evaluation methods were non-reference. Also, there have been studies that specifically propose evaluation frameworks for DVS systems [103][12][104][105][106][107][108][109][110]. However, these frameworks consider non-reference evaluations, not taking advantage of the availability of GT videos. Finally, there have been a few full-reference evaluation proposals in [111][97]. Both works present complex metrics, however fail to provide an evaluation code, which could lead developers to obtain wrong calculations.

2.5 Image alignment

Image alignment is one of the main components of our work. Hence, in this Section we present a number of proposed systems for homography estimation, image registration, and visual odometry estimation using different strategies, both traditional and DL-based. This Section also presents a number of proposals that utilize STNs in their architectures.

2.5.1 Homography Estimation

Chang et al. [112] propose a network architecture that combines the strengths of CNNs and the Lucas-Kanade algorithm for image alignment. The architecture contains a Lucas-Kanade layer that performs the inverse compositional algorithm on convolutional feature maps. The method contains a cascaded feature learning method that incorporates

the coarse-to-fine strategy into the training process, learning a pyramid representation of convolutional features in a cascaded manner and yielding a cascaded network that performs coarse-to-fine alignment on the feature pyramids. The method is then applied to the task of homography estimation.

DeTone et al. [113] present HomographyNet, a CNN for estimating the relative homography between image pairs. The proposed network takes two stacked grayscale images as input, and produces an 8 DoF homography matrix. The work presents two architectures for HomographyNet: 1) a regression network to estimate the real-valued homography parameters, and 2) a classification network to produce a distribution over quantized homographies. To this end, a 4-point homography parameterization is utilized, maping the four corners from one image into the second image.

Nowruzi et al. [114] propose a hierarchy of twin convolutional regression networks to estimate the homography between image pairs. The proposed approach contains sequentially stacked networks to reduce error bounds of the estimate. At every convolutional network module, features from each image are extracted independently, given a shared set of kernels, in a siamese network. Later on in the process, they are merged together to estimate the homography. Given the iterative nature of the framework, highly complicated models are not necessarily required, and high performance is achieved via hierarchical arrangement of simple models.

2.5.2 Image Registration

Fan et al. [115] propose an unsupervised adversarial similarity network for image registration. The given approach does not require ground-truth deformations and specific similarity metrics. Instead, a registration network and a discrimination network are connected with a deformable transformation layer. With adversarial training, the registration network is trained to predict deformations that are accurate enough.

Huang et al. [116] propose a combination of unsupervised joint alignment with unsupervised feature learning. Specifically, it incorporates deep learning into the alignment framework, which iteratively transforms images to reduce the entropy of a given dataset. The proposed approach obtains features that can represent the image at differing resolutions based on network depth, and that are tuned to the statistics of the specific data being aligned.

Kowalski et al. [117] propose Deep Alignment Network (DAN), a robust face alignment method based on a deep neural network architecture. DAN contains a number of stages, each stage improving the locations of the facial landmarks estimated by the previous stage. The proposed method uses entire face images at all stages, and allows DAN to handle face images with large variation in head pose and difficult initializations.

2.5.3 Visual Odometry estimation

Constante et al. [118] propose a novel frame-to-frame (F2F) Visual Odometry (VO) strategy using a CNN. The approach can autonomously select the most important visual cues and the best strategy to compute F2F estimates that are robust to blur, luminance and constrast anomalies.

DeepVO [119], proposed by Wang et al., is an end-to-end framework for monocular VO that uses deep Recurrent Convolutional Neural Networks (RCNNs). It infers poses directly from raw videos without adopting any module in the conventional VO pipeline. DeepVO automatically learns effective feature representation for the VO problem through CNNs and implicitly models sequential dynamics and relations using deep Recurrent Neural Networks.

Yang et al. [120] propose to leverage deep monocular depth prediction to overcome limitations of geometry-based monocular visual odometry. The proposed system

incorporates deep depth predictions into Direct Sparse Odometry (DSO) as direct virtual stereo measurements. The depth prediction module consists of a deep network that refines predicted depth from a single image in a two-stage process: the network is trained in a semi-supervised way on photoconsistency in stereo images and on consistency with accurate sparse depth reconstructions from stereo DSO.

Iyer et al. [121] propose an unsupervised paradigm for deep visual odometry learning. The authors design a loss term that enforces geometric consistency of the trajectory which allows accurate training of unsupervised VO models. The proposed method uses geometry as a self-supervisory signal and Composite Transformation Constraints (CTCs), that automatically generate supervisory signals for training and enforce geometric consistency in the VO estimation.

2.5.4 Spatial Transformer Networks

Cirstea et al. [122] propose Tied Spatial Transformer Networks (TSTNs), a combination of CNNs and STNs which consists of a localization CNN and a classification CNN with shared weights. The former predicts an affine transform for the input image, which is processed according to the predicted parameters and passed through the classification CNN. The authors then utilize the approach to the noisy digits recognition, using the cluttered MNIST database [123], which consists of the MNIST [124] database with additional noise.

Kim et al. [125] present recurrent transformer networks (RTNs), a group of networks for obtaining dense correspondences between semantically similar images. The work consists of an iterative process that estimates spatial transformations between the input images and generates aligned convolutional activations with the estimated transformations. Instead of using STNs to independently normalize each individual image, the system directly estimates the transformations between image pairs. The process is

performed recursively to refine the transformation estimates and the feature representations. Finally, the authors present a technique for weakly-supervised training RTNs that is based on a classification loss.

Lin et al. [43] establish a theoretical connection between the classical Lucas-Kanade (LK) algorithm and STNs. The authors present Inverse Compositional Spatial Transformer Networks (IC-STNs), inspired by the Inverse Compositional (IC) variant of the LK algorithm. Then, they demonstrate that IC-STNs achieves better performance than conventional STNs with less model capacity, in the context of image alignment tasks and joint alignment and classification problems.

Lin et al. [41] propose Spatial Transformer GANs (ST-GANs), a GAN architecture that utilizes STNs as the generator to find geometric corrections to a foreground object that is composited into a background image. The proposed approach exploits an iterative STN warping scheme and propose a sequential training strategy that achieves better results compared to naive training of a single generator.

Garcia et al. [126] present a Deep Learning approach for traffic sign recognition systems which comprises Convolutional layers and Spatial Transformer Networks. The authors perform a number of experiments to measure the impact of diverse factors aiming at designing a CNN that improves the state-of-the-art. First, a number of different adaptive and non-adaptive Stochastic Gradient Descent (SGD) optimisation algorithms (such as SGD, SGD-Nesterov, RMSprop and Adam) are evaluated. Then, multiple combinations of STNs placed at different locations within the neural network are analysed.

Fang et al. [127] propose a Multi-Level Feature Fusion Based Locality-Constrained Spatial Transformer Network (MLSTN), which consists of a density map regression module and a Locality-Constrained Spatial Transformer (LST) module. It first estimates

the density map of each frame by utilizing the combination of the low, middle and high level features of the CNNs. To measure the relationship of the density maps between neighboring frames, the LST module estimates the density map of the next frame by concatenating several regression density maps.

2.6 Filters

We use Motion Vector Integration and Frame Position Smoothing in this thesis. Hence, in this Section, we provide background on a number of filters (Kalman and low-pass) that we are going to utilize.

2.6.1 Butterworth Low-Pass Filter

The Butterworth low-pass filter is commonly used in gait analysis applications. It is widely utilized by the community, due to its simplicity and acceptable performance. The only parameter to be defined by the user is the cutoff frequency. Such filter operates on the assumption that the signal to be processed is stationary, which is not always the case. The performance of the Butterworth filter can, therefore, be limited by such fact [128].

This filter is considered the best compromise between attenuation and phase response, since it presents no ripple in the pass band or the stop band. Due to this reason, it is also known as maximally flat filter. However, the Butterworth filter achieves its flatness at the expense of a relatively wide transition region from pass band to stop band [3]. A frequency response chart to the Butterworh filter is shown in Figure 2.5.

We chose the Butterworth filter as an option for video stabilization because the one of the main sources of jitter in amateur videos is the movement of a walking person.

Further in this thesis, we analyse the performance of this filter under a number of cutoff frequencies, to observe whether it is suitable for video stabilization.

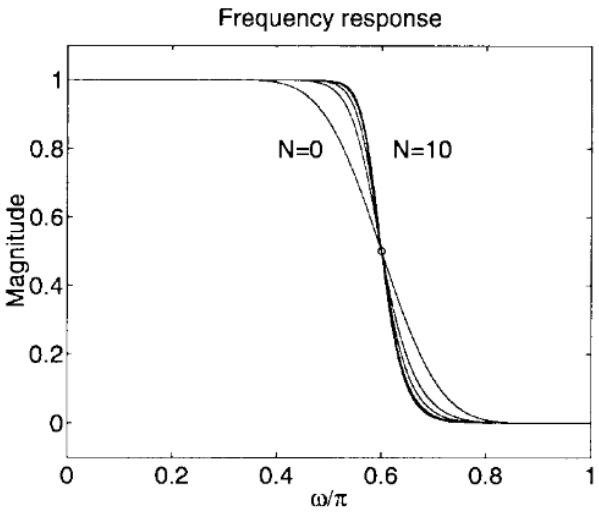


Fig. 2.5 Sample frequency response of the Butterworth filter (extracted from [2]).

2.6.2 Gaussian Low-Pass Filter

Gaussian kernel filtering is a widely adopted method to remove high frequency jitter from videos because it is simple to implement and to adjust the level of stabilization. It avoids overshoot to a step function input while minimizing rise and fall time [27], as can be seen in Figure 2.6. As it is widely known, its filtering characteristics rely on the variance of the gaussian curve, which allows for the adjustment of the level of stabilization [9].

The Gaussian filter is a popular approach to smoothing the camera path in traditional DVS systems. Therefore, we take it into account when analyzing low-pass filters further in this thesis.

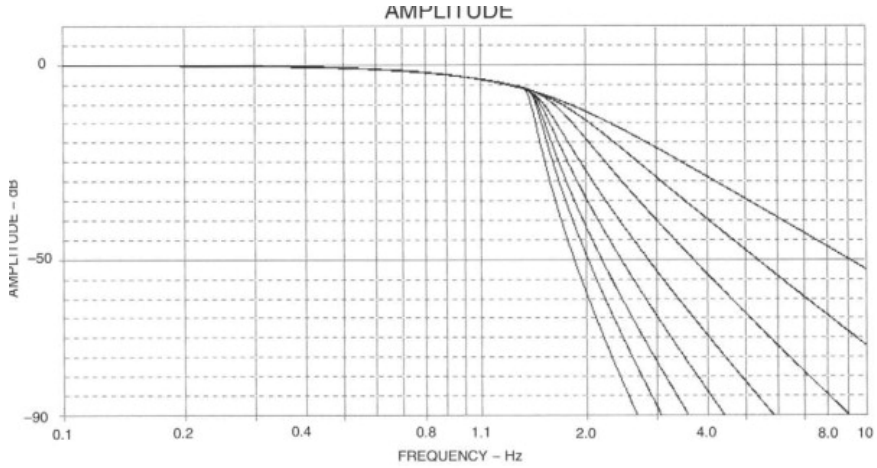


Fig. 2.6 Sample frequency response of the Gaussian filter (extracted from [3]).

2.6.3 Kalman Filter

The Kalman filter uses the camera path to separate wanted from unwanted motion, correcting the latter [9]. It is a set of mathematical equations that provide an efficient computational and recursive solution of the least-squares method. The filter supports estimations of past, present, and future states. The Kalman filter has been extensively utilized in the academia and industry, especially in autonomous or assisted navigation [129]. Although we do not intend to get into details, it heavily relies on the prediction error covariance of the process it tries to filter.

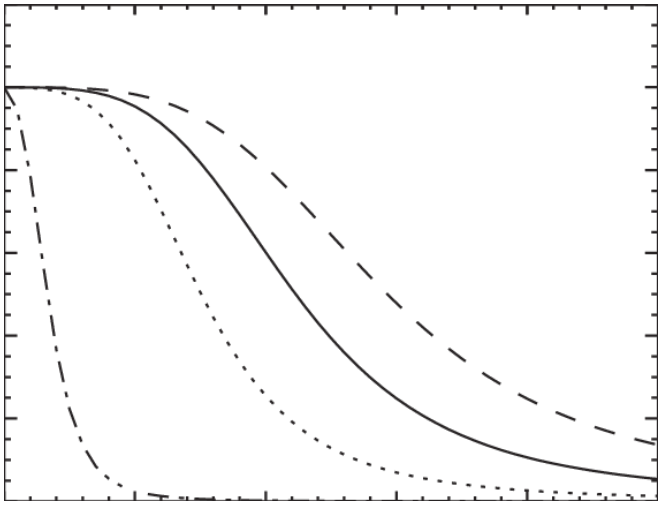


Fig. 2.7 Sample frequency response of the Kalman filter (extracted from [4]).

The frequency response of the Kalman filter is shown in Figure 2.7. There is a number of DVS methods that employ this filter in their strategies. Hence, we consider it a good filter to consider when it comes to assessing low-pass filters for DVS.

2.7 Proposal

To address the shortage of DL-suitable datasets, in this thesis we provide a dataset of synchronised stable-unstable video pairs with the motion parameters that maps them. To the best of our knowledge, our dataset is the largest ever provided for DVS systems, and is the only one to provide motion parameters between ground truth and unstable videos. Since our video production is synthetic, our dataset can scale easily, and does not require time consuming recordings.

With our dataset, we provide a full-reference performance evaluation framework, to help the development and improvement of DVS systems. Since we provide a dataset that contains motion parameters, our evaluation framework also provides metrics that take into account these parameters. We believe these tools will contribute to the development of future DL-based DVS systems.

Traditional and Deep DVS methods present strengths and weaknesses. Although traditional DVS methods can effectively smooth camera path, they struggle to extract and track features in a number of scenarios. Deep Learning-based video stabilization proposals, on the other hand, are effective in feature extraction and motion parameter estimation. However, they struggle with camera path smoothing, since it is a hard problem to define in this context. Also, current DL-based DVS systems tend to add artifacts to the resulting videos, such as distortion and blur, by either performing 2D transformations or cascaded interpolation to the unstable videos. These artifacts tend may be more annoying than the original jitter of the unstable video.

Having this in mind, we propose our novel STN-based video stabilization system. We first describe our image alignment module. Such module simplifies even further the training process, since it doesn't require stable/unstable video pairs, but images, which allows the utilization of image datasets. Then, we propose a series of DVS methods. The first one consists of a homography-based video stabilization. The second is a translation-based video stabilization that leverages EWMAs and moving averages to smooth the camera path. Lastly, we present a filter-based DVS. In the latter proposal, we present a thorough analysis of filters, such as Gaussian, Kalman, Butterworth, amongst others. We then select the filters and parameters that present the best performances for both the homography-based and the translation-based systems. We then compare the performance of our three proposed DVS systems, and determine which one is the best amongst them. Finally, we study the possibility of adding our system to StabNet, our Deep-based baseline.

Our proposals are effective, since they are not sensitive to scene content, and simplify the camera path smoothing problem. Furthermore, our final systems avoid blurry and distorted videos, since we simulate translation only, which we prove to be effective in our experimental results. This allows us to overcome the issue presented by current DL-based DVS systems, and is an initial step in building effective DVS systems that avoid artifacts in their stabilized videos. Added to these, our in-depth filter selection shows that there is room for improving current filters utilized for video stabilization.

Our proposed systems are different from our DL-based baseline, StabNet [23], which uses a siamese network to obtain a grid of motion parameters, enforcing temporal and spatial constraints to obtain a stable video. Its entire DVS implementation is based on Deep Learning, and the camera path is smoothed considering a number of previous frames: for each frame, these previous frames are taken into account to estimate a grid of motion parameters (see Figure 2.8). Since StabNet estimates a grid of transformation

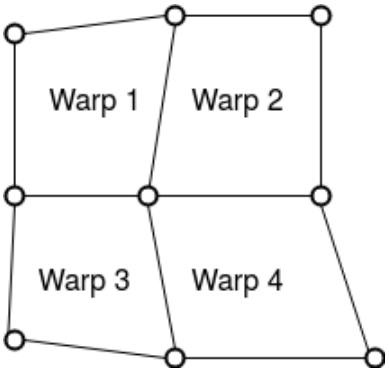


Fig. 2.8 StabNet warp grid.

parameters, it needs to reinforce inter-grid consistency between all sub-images. It leads to a small output frame, and in some cases the consistency enforced is not strong enough to avoid distortions between sub-images. Added to this artifact, for each sub-image, StabNet performs a homography, which adds distortion and blur to the sub-image.

Our systems, on the other hand, consist of a mixture of Deep Learning and traditional methods. Our motion estimation is Deep Learning-based, while the camera path smoothing is traditional-based. Unlike StabNet, our systems estimate global motion parameters, which mitigates intra-frame distortions caused by sub-image stitching.

However, in the homography-based system, one can see the distortion and blur caused by the homography, as well as unnatural motion in case there is occlusion, or parallax. With our translation-based system, on the other hand, since we do not warp the frames, we avoid distortion and blur. We also avoid the unnatural view previously seen with our homography-based system. Hence, we argue that our system is the best option so far.

Chapter 3

Dataset and Evaluation Framework

3.1 Motivation

The most adequate technique for casual amateur recording is Digital Video Stabilization (DVS), which is convenient and economical [13]. The literature has reported that Deep Learning (DL) approaches have demonstrated the ability to handle and process complex, large-scale datasets for addressing various computer vision challenges, such as super resolution [18], image deblurring [19], style transfer [20], amongst others. In this scenario, DL-based approaches for DVS have recently been proposed [23][24][25][26][96]. However, large datasets are essential for learning-based algorithms: for DVS, such datasets would normally require pairs of synchronized steady-unsteady videos [23][24]. In fact, Wang et. al. [23] and Xu et. al. [24] point out that the lack of appropriate datasets is one of the root causes for the few DL-based DVS systems available to date, and [97] and [23] provide datasets for DVS. However, they consist of short amounts of video pairs and do not provide motion parameters that map steady and unsteady frames.

Having in mind such shortage and the fact that current DL-based DVS systems estimate transformation matrices in their algorithms, in this Chapter we provide a dataset that contains such information. We leverage the knowledge provided by the dataset in [23] to generate a large dataset of artificially produced steady-unsteady pairs. Unlike [23], we provide motion parameters, which can decrease the complexity in the training process of future algorithms, since the loss function can be based on the motion parameters instead of the output frames.

Also, performance evaluation in previous studies have mostly been non-reference (i.e., taking only the stabilized video into account). Since we provide a large amount of Ground Truth (GT) and parameter (PR) data, we implement a full-reference evaluation framework, for performance evaluation based on GT, PR, and stabilized data. We have proposed such a framework to simplify the evaluation process, since using the GT allows the use of efficient and simple metrics such as Mean Square Error (MSE), Structural Similarity Index (SSIM), and to directly measure how much a feature has moved between GT and stabilized video. Also, by using PR information we allow an even simpler comparison, between estimated and ground truth motion parameters.

To demonstrate the practical use of the introduced dataset and the evaluation framework, we compare the performances of Estadeo [28], a 2D-based DVS algorithm, and StabNet [23], a DL-based DVS algorithm. The contribution[1] of the Chapter is threefold: i) a dataset for training and testing DL-based DVS systems, which consists of a variety of video types; ii) a full-reference evaluation framework, which considers GT frames and motion parameters; iii) with our newly proposed evaluation framework, an initial performance comparison between a traditional and a DL-based DVS systems.

[1] Dataset and framework available at https://github.com/mariito/DVS_

3.2 Dataset Production

This Section describes our dataset production. As [23][24] mentioned, one of the main obstacles for developing DL based Video Stabilization is the lack of training dataset. The authors in [23] and [97] have manually recorded their datasets, since they utilized a device for producing steady and unsteady video pairs: they had two cameras attached to a handheld stabilizer, next to each other. The first one was fixed to the stabilizer, producing a video with motion similar to that of the holder of the stabilizer. The second one was attached to a moving platform in the stabilizer, producing stable videos. Although it is a feasible way to produce a dataset, it demands recording of all the videos in the dataset. It is clear that producing large datasets is a time consuming task with this approach. In fact, the amount of video pairs provided is low: around 105 for both datasets, and in [23] the authors utilized data augmentation techniques to train their system.

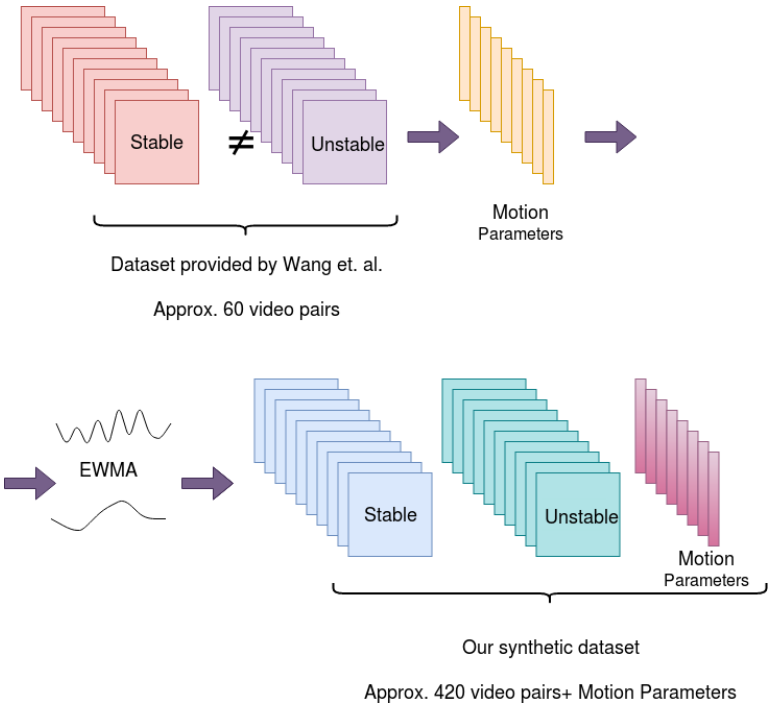


Fig. 3.1 Dataset Creation Procedure.

Algorithm 1 Unsteady Video Production. V_{St} and V_{Un} are the stable and unstable videos from the dataset in [23]. $frames$ and fr_{Uns} are the frames of stable and unstable videos, respectively. $Transform$ is the list of transformation parameters estimated between V_{St} and V_{Un}. V_{Orig} is the stable video that is going to be used to generate an unsteady video. $Smooth_{TR}$ is the smooth transformation list, obtained by applying an EWMA to $Transform$. fr_{syn} is the frame that has been synthetically transformed, to generate an unsteady video V_{Syn}.

procedure UNSTEADYVIDEO(V_{St}, V_{Un}, V_{orig}) ▷ Input:V_{St}, V_{Un}, V_{orig}
  Assert $length(V_{St}) = length(V_{Un})$
  for Vid in V_{St} **do**
    Read $frames(Vid, Vid_{Uns})$
    for fr in $frames$ **do**
      Read fr and fr_{Uns}
      $R_{Tr} \leftarrow EstimateRigidTransform(fr, fr_{Uns})$
      Append R_{Tr} to $Transform$
    end for
    Save $Transform$
  end for
  for Vid in V_{Orig} **do**
    Read $frames(Vid)$
    for fr in $frames$ **do**
      Pick R_{Tr} from $Transform$
      $Smooth_{TR} \leftarrow EWMA(previousframes)$
      $fr_{syn} \leftarrow warp(fr, Smooth_{TR})$
      Save fr_{syn}
      Append fr_{syn} to V_{Syn}
    end for
    Save V_{Syn}
  end for ▷ Output:V_{Syn}, $Smooth_{TR}$
end procedure

With this in mind, our approach to providing a dataset is by producing synthetic stable-unstable video pairs and motion parameters, as seen in Figure 3.1 and presented in Algorithm 1. Since our video production is synthetic, our dataset can scale easily, and does not require manual recordings.

First, we collect the dataset provided by [23], then we extract the frames from each video pair (using ffmpeg [130]) to compare their frames. We estimate a 2x3 transformation matrix between each frame pair. For some of the pairs, we can not estimate such matrix due to the content of the frames, however the amount of transformation matrices obtained (approximately 23000 records) is considered a good sample for our purposes.

We collect short videos from a free stock video repository[2], which contains several types of high quality and steady videos. We classify them into 9 categories: a) Simple, videos with the same depth and textured objects; b) Blurry; c) High motion; d) Dark; e) Textureless; f) Parallax; g) Discontinuous depth; h) Crowd, videos with large amounts of moving objects, with high motion and parallax; i) Close object, videos with at least one close object, leading to obstruction. The Simple category was selected because it contains features that have been previously addressed by DVS systems, and should not pose a problem to video stabilization algorithms. The remaining video categories were selected because they contain one characteristic that poses a challenge to current DVS systems.

From this group of steady videos, we utilize the transformation matrices we obtained from [23] to produce our dataset of unsteady videos: for each frame in a given video, we randomly select one affine matrix and assign it to the given frame. To avoid a wobbly unnatural video, we utilize an Exponential Weighted Moving Average (EWMA) to smooth the transition between frames, then crop the unstable frames, to extract the

[2]https://www.pexels.com/

black borders resulting from the warps. To maintain resolution consistency between GT and unstable videos, we also crop the GT frames, so both videos present the same resolution. Then, we produce videos with the frames using ffmpeg [130].

We perform all these steps to make the motion in the unstable videos as close to real unstable videos as possible. On the other hand, we also aim at producing a wide variety of jitter frequencies: for producing our dataset, we do not take into account the motion patterns of the videos in [23], only the motion parameters. Algorithm 1 shows an overview of the unstable video production.

3.2.1 Dataset Characteristics

Table 3.1 and Figure 3.2 present the characteristics of the videos collected for our dataset. They consist of 421 videos, with a variety of durations, resolutions (presented separately as width and height), bitrates, and frame rates. We present the mean and standard deviation for all the videos (All), and for each video category, separately (Simple, Blurry, High Motion, Dark, Textureless, Parallax, Depth, Crowd, and Close Object). Most videos would fit in more than one category, therefore we added each video to the most challenging category it would be part of. Hence, the Textureless, Depth, and Close Object categories contain more videos.

The duration for most categories are around 10s and present certain consistency, with a high standard deviation in the Parallax, Depth, Crowd, and Close Object categories. The bitrate, however, presents the most significant changes within the same category, which can be explained by the relationship between bitrate and video content. It is also interesting to note that the Blurry and Close Object categories present the lowest bitrate, whereas the High Motion presents the highest ones. It is expected, since blurry scenes and close objects present a high amount of patterns, whereas high motion presents a wide variety of changes in scene content. Lastly, the width, height, and frame rate

present certain consistency between categories and within the same category. It is caused by the tendency of the video content providers to maintain high quality of the content they provide.

	Amount		Duration (s)	Bitrate (kbps)	Width	Height	Frame Rate (fps)
All	421	AVG	11.59	9,767.07	2,504.97	1,448.21	28.14
		STD	6.28	10,099.69	1,023.35	574.82	7.33
Simple	62	AVG	9.56	12,124.00	2,572.29	1,430.65	27.88
		STD	1.45	11,339.76	1,078.52	584.68	6.55
Blurry	23	AVG	9.34	6,943.70	2,265.04	1,267.83	26.43
		STD	1.29	8,358.07	794.16	432.40	2.70
High Motion	43	AVG	9.73	13,318.63	2,836.09	1,710.70	28.01
		STD	1.01	11,774.23	993.93	723.37	7.62
Dark	43	AVG	9.34	12,999.00	2,878.93	1,727.40	28.91
		STD	1.49	13,950.11	1,067.59	499.53	9.07
Textureless	96	AVG	9.80	10,012.70	2,729.13	1,581.79	27.53
		STD	0.87	9,834.66	1,048.72	551.29	7.68
Parallax	19	AVG	12.32	10,898.05	2,694.95	1,514.11	28.51
		STD	5.59	9,988.16	1,049.64	571.61	5.85
Depth	73	AVG	17.87	7,832.41	2,301.12	1,313.95	28.97
		STD	9.48	6,897.58	851.41	452.64	8.36
Crowd	10	AVG	12.75	7,708.00	2,265.60	1,260.00	28.38
		STD	5.01	6,757.42	921.15	487.44	2.58
Close Object	52	AVG	12.43	4,841.48	1,796.77	1,055.04	28.49
		STD	9.85	5,454.73	816.24	466.97	6.86

Table 3.1 Dataset Characteristics: Original Videos.

We use the videos in Table 3.1 to produce our Video Stabilization Dataset. Table 3.2 presents the characteristics of our synthetic dataset. This Table only shows the unstable videos and does not contain the frame rate column, since all videos have been produced with the same frame rate: 30 fps. We also do not present duration column, since they are very similar to the original videos. It is expected that the width, height, and bitrates are slightly lower than the original videos, since we have cropped all GT and unstable videos.

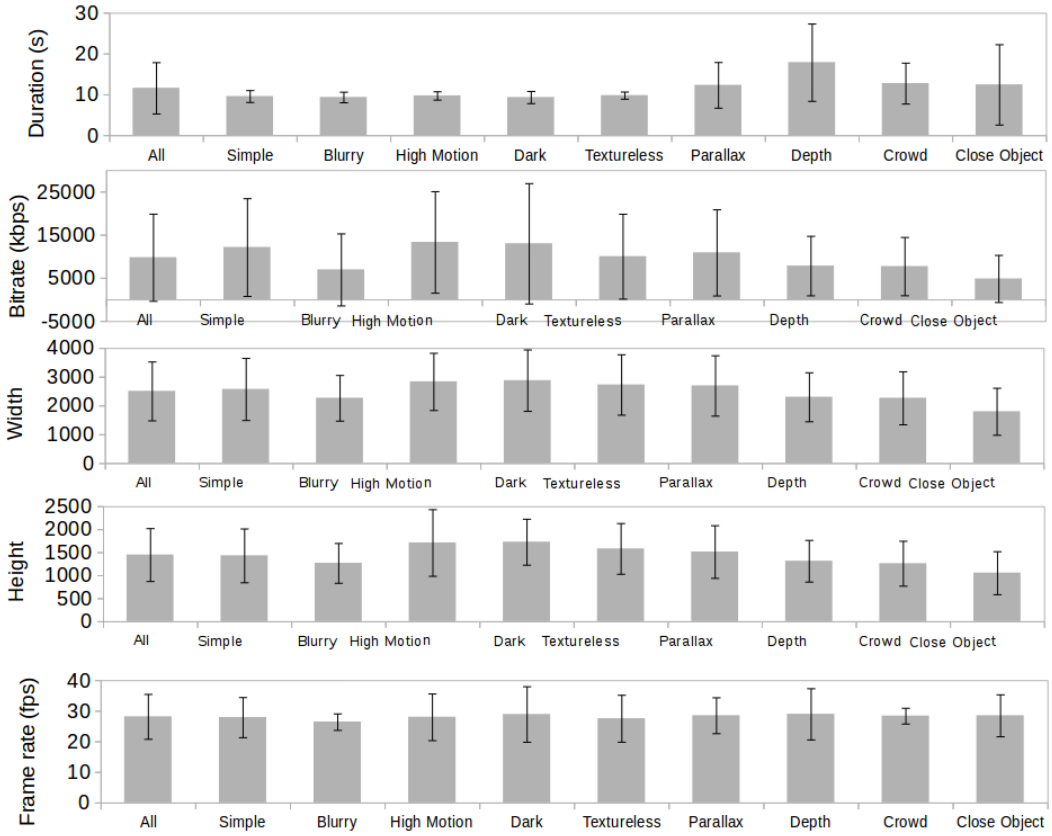


Fig. 3.2 Dataset Characteristics: Original Videos.

3.3 Other Datasets Utilized in This Thesis

The following datasets are also present in this thesis.

COCO Dataset [5]

A large scale dataset (see Figure 3.3) that addresses three main research problems in scene understanding: detecting non-iconic views (or non-canonical perspectives) of objects, contextual reasoning between objects and the precise 2D object location. The authors argue that current recognition systems perform fairly well on iconic views (i.e., images that contain the object with no obstructions next to the center of a neatly composed photo), but struggle to recognize objects otherwise (in the background,

	Amount		Width	Height	Bitrate GT (kbps)	Bitrate Unstable (kbps)
All	421	AVG	2,254.47	1,303.39	9,181.05	8,399.68
		STD	921.02	517.34	9,089.72	8,584.74
Simple	62	AVG	2,315.06	1,287.59	10,911.60	10,790.36
		STD	970.67	526.21	10,319.18	9,752.19
Blurry	23	AVG	2,038.54	1,141.05	6,596.52	6,041.02
		STD	714.74	389.16	7,689.42	7,187.94
High Motion	43	AVG	2,552.48	1,539.63	12,519.51	11,853.58
		STD	894.54	651.03	11,067.78	10,008.10
Dark	43	AVG	2,591.04	1,554.66	11,829.09	11,179.14
		STD	960.83	449.58	12,555.10	12,136.60
Textureless	96	AVG	2,456.22	1,423.61	9,211.68	8,610.92
		STD	943.85	496.16	9,342.93	8,851.19
Parallax	19	AVG	2,425.46	1,362.70	10,244.17	9,263.34
		STD	944.68	514.45	9,189.11	8,889.46
Depth	73	AVG	2,071.01	1,182.56	7,049.17	6,970.84
		STD	766.27	407.38	6,276.80	6,069.87
Crowd	10	AVG	2,039.04	1,134.00	7,399.68	6,705.96
		STD	829.04	438.70	6,419.55	5,743.81
Close Object	52	AVG	1,617.09	949.54	4,647.82	4,212.09
		STD	734.62	420.27	4,963.80	4,909.26

Table 3.2 Dataset Characteristics: Ground Truth and Unstable Videos.

partially occluded, or amid clutter). In fact, finding natural images that contain several objects (i.e., images depicting scenes instead of isolated objects) is challenging.

The Microsoft Common Objects in COntext (MSCOCO) dataset consists of 91 common object categories, with 82 of them having more than 5,000 labeled instances. The dataset has a total of 2,500,000 labeled instances in 328,000 images. Unlike the ImageNet dataset [131], the COCO dataset has fewer categories but more instances per category. This is helpful in learning detailed object models capable of accurate 2D localization. Another distinction between the COCO and other datasets is the number of labeled instances per image, which may aid in learning contextual information.

DeepStab Dataset [23]

A dataset provided by Wang et. al. containing pairs of synchronized videos with diverse scenarios: indoor scenes with parallax, and outdoor scenes with buildings, crowds,

Fig. 3.3 COCO train dataset sample [5].

vegetation, amongst others. The camera paths present in this dataset is also varied: moving camera, panning, and combinations of both, at various speeds. The dataset contains 60 video pairs of synchronized videos with length of up to 30s, at 30 fps.

3.4 Evaluation Framework

In the literature, there have been few studies dedicated to evaluating the perceptual quality of stabilized videos [132][12][111][106][133][103]. However, there is no widely accepted evaluation framework in the academia, since it is hard to quantify the stability of a video. On the other hand, although it is possible to perform subjective tests on the output of DVS systems, these tests are hard to perform and time consuming.

Although objective metrics do not thoroughly assess the quality of stabilized videos, they are more straightforward than subjective tests and tend to show an overall view on the DVS systems. Full-Reference (FR) metrics compare the stabilized videos and the ground truth ones, whereas Non-Reference (NR) directly evaluate the stabilized videos. However, there has been little progress in performance evaluation for DVS [97][9], with a majority of performance metrics being NR, since most available datasets consist

of only the unsteady videos. With our new dataset, we provide a FR performance evaluation framework.

This Section presents the metrics and formulae of our evaluation framework. It is FR, since we provide a wide range of videos for tests, with the ground truth and motion parameters. Consider a stable-unstable video pair, which consist of n frames $S_{gt} = \{F_1, F_2, ..., F_n\}$ and $S_{un} = \{\hat{F}_1, \hat{F}_2, ...\hat{F}_n\}$, respectively. Also, consider a stabilized video, which is the output of a DVS system and consists of n frames $S_{st} = \{\bar{F}_1, \bar{F}_2, ..., \bar{F}_n\}$. The metrics we evaluate in our framework compare stabilized S_{st} and GT frames S_{gt}, and consist of:

i) Mean Square Error (MSE), Eq.3.1: the mean MSE between $\bar{F}_i$ and F_i frames of a video.

$$MSE = \frac{1}{n} * \sum_{i=1}^{n} MSE_i(\bar{F}_i, F_i) \tag{3.1}$$

ii) Structural Similarity Index (SSIM), Eq.3.2: the mean SSIM between $\bar{F}_i$ and F_i frames of a video.

$$SSIM = \frac{1}{n} * \sum_{i=1}^{n} SSIM_i(\bar{F}_i, F_i) \tag{3.2}$$

Since these metrics compare $\bar{F}_i$ and F_i, they are computing all types of distortion (noise, blur, the distortion of straight lines, among others) in the resulting frame.

iii) Distance between features: shows how much a given feature has moved from GT to stabilized frame. Consider a given set of features, which has coordinates $(x_{gt}, y_{gt})_1, (x_{gt}, y_{gt})_2, ..., (x_{gt}, y_{gt})_n$ in the GT frame and $(x_{st}, y_{st})_1, (x_{st}, y_{st})_2, ..., (x_{st}, y_{st})_n$ in the stabilized frame. Both stabilized and ground truth frames have the same dimen-

sion, y_{fr} width and x_{fr} height. The distance D_f between features can be computed as shown in Eq.3.3 and Eq.3.4. The notion behind this is that, on average, the higher the mean distance between features in a frame pair, the more the stabilized frames are dislocated.

$$D_f(x) = \frac{1}{n * x_{fr}} * \sum_{i=1}^{n} |x_{st} - x_{gt}|_i \tag{3.3}$$

$$D_f(y) = \frac{1}{n * y_{fr}} * \sum_{i=1}^{n} |y_{st} - y_{gt}|_i \tag{3.4}$$

iv) Resolution Preservation (ResPrev), Eq.3.5: compares the average ratio of file size (F_{st} and F_{gt}) and number of pixels (P_{st} and P_{gt}) between S_{st} and S_{gt}, respectively. This metric aims at determining how much the video has been cropped, and if there has been any frame quality loss.

$$R_P = \frac{1}{n} * \sum_{i=1}^{n} \frac{(F_{st}/P_{st})_i}{(F_{gt}/P_{gt})_i} \tag{3.5}$$

v) Final resolution of the outputs to the DVS systems, given the same input size (Eq. 3.6.

$$F_R = y_{fr} * x_{fr} \tag{3.6}$$

The previous state-of-the-art papers propose slightly different metrics. The non-reference ones are not needed in our framework, since we can perform a full-reference evaluation. Some of the full-reference metrics will be added to our framework in

Fig. 3.4 Videos utilized in our performance evaluation

future work. We validate our evaluation framework in Section 3.5 by comparing two state-of-the-art DVS methods.

3.5 Experiments

In this Section, we evaluate two selected state-of-the-art proposals on DVS using the videos in Figure 3.4. We choose these proposals because their authors have made their code available to the public:

i) Estadeo [28]: is a paper dedicated to implementing and exhaustively comparing classic DVS techniques and boundary conditions. In this Chapter, we will utilize the default options in the code provided by the author.

ii) StabNet [23]: is a low-latency, real-time, DL-based method. It learns a set of transformations for each input frame, considering the previously stabilized frames in

the video. In this Chapter, we will utilize the code and pre-trained model provided by the authors.

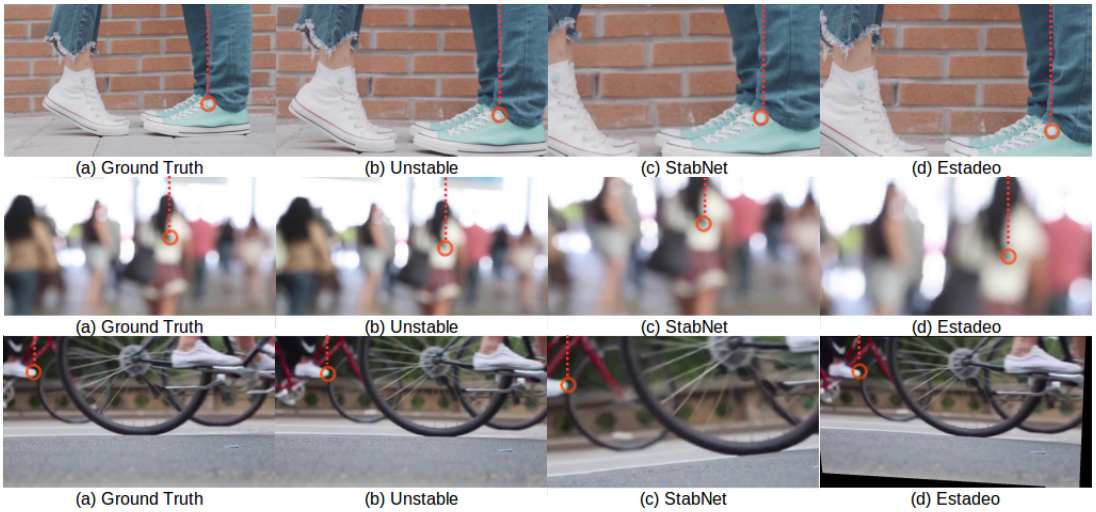


Fig. 3.5 Sample frames for videos a) (Simple), b) (Blurry), and c) (High Motion).

We encourage the reader to refer to Chapter 2 for a thorough description of StabNet and Estadeo. By following the authors' instructions, we tend to be fair to both systems, which is our main goal in our performance evaluation. Figure 3.5 shows a randomly selected sample of (from left to right) the stable frame, unstable frame, and the outputs to StabNet and Estadeo, respectively. In the Figure, we highlight how much a given feature has moved from GT to unstable and output frames.

Figure 3.6 shows the charts with the performance evaluation we executed with our framework. The blue bars represent the results we obtained for StabNet, whereas the orange bars represent the results for Estadeo. The x-axis represents the 9 different videos, shown in Figure 3.4: 1-9 correspond to videos (a) to (i), respectively. In our evaluation, we cropped the GT frames to have the same dimensions as the outputs to the systems, to calculate metrics such as MSE.

By observing Figure 3.6, we note that both systems present high MSE, and it depends on the nature of the video. Most MSE values are similar between the systems, and

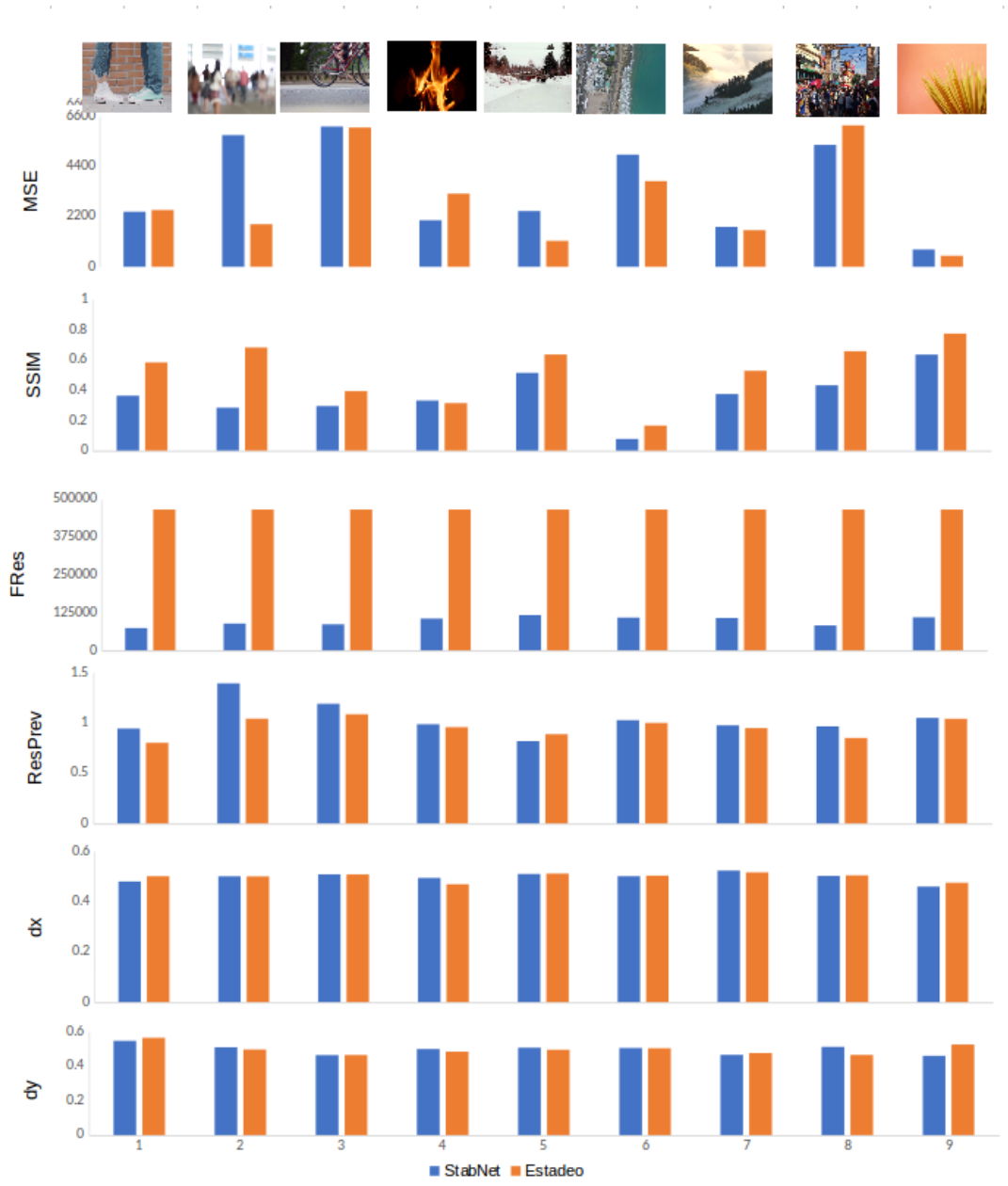


Fig. 3.6 Experimental results. The blue bars represent StabNet, the orange bars represent Estadeo. Experimental results: 1-9 in the x-axis correspond to videos (a)-(i).

when it differs significantly, it is mainly with StabNet presenting worse results. The output to Estadeo presents better SSIM results in most cases, and both systems have very low performance in video 6 (parallax). The Resolution Preservation is similar for both systems in most cases, and when they differ significantly, Estadeo has the worst result. It means that, although Estadeo tends to present better MSE and SSIM

results and has significantly higher output resolution, it presents worse performance when it comes to the video quality, which could be annoying to the viewer of the video, or could impair the performance of a given computer vision algorithm. However, the ResPrev results are probably a cause of the high final resolution, which preserves the initial resolution at the cost of lower video quality.

Finally, dx and dy, which show normalized distance between feature coordinates in the GT and stabilized frames, demonstrate that the features move at similar average distances in the x and y-axis for both Estadeo and StabNet.

Although both systems present similar performances, Estadeo still has an advantage when compared to StabNet, if we consider carefully MSE, SSIM, and Final Resolution. Even though the Resolution Preservation results are not in favour of Estadeo, one could argue that it can be frustrating to view an output video with significantly lower resolution, even though it presents higher resolution preservation. Hence, we consider that Estadeo outperforms StabNet in our initial evaluation.

3.6 Other Metric Utilized in This Thesis

Video Multimethod Assessment Fusion (VMAF) [134]

VMAF combines a number of video quality features: 1) Detail Loss Metric (DLM [135]), 2) Visual Information Fidelity (VIF [136]), and 3) luminance differences between frame pairs. 1) and 2) are metrics based on the Human Visual System, whereas 3) adds the temporal information. These features are combined using a Support Vector Regression (SVR) trained on subjective data. Since VMAF is trained on subjective data, we argue that there is a subjective factor to this metric.

Although VMAF does not fully exploit temporal quality information sensitive to temporal distortions, it focuses on compression and scaling artifacts. The former tipically consists of blocky regions within a frame, and the latter is observed as blurry regions around edges. Since it exploits loss of information and blurriness in frames, we argue that it can be used as a perceptual video metric to evaluate blur in the stabilized videos.

Since VMAF is Full-Reference, it compares a distorted image to a reference, given that both show exactly the same scene. It ranges between 0 and 100, and for VMAF scores above 96, the distorted video is considered perceptually inseparable from the reference. In our case, the videos are not the same, since we are comparing a stable video (reference) to a stabilized one (distorted). It is expected that the VMAF scores are going to be low: in fact, the more misaligned two frames are, the lower the VMAF score, hence we argue that it indirectly measures the alignment between two videos. Since the reference is a stable video, it indirectly measures how stable a stabilized video is.

3.7 Outcomes

This Chapter has provided fundamental tools for the development of Deep Learning-based Video Stabilization Systems. The first is a comprehensive synthetic dataset, which consists of Ground Truth and unsteady frames, and the motion parameters that map each frame of the stable video into the corresponding frame in the unstable video. We also presented a full-reference performance evaluation for DVS algorithms, which takes into account the provided features in the dataset. Then, we compared the performances of two state-of-the-art DVS systems: a 2D based method, Estadeo, and a DL-based approach, StabNet. Both systems showed similar performance, although Estadeo performed better in some scenarios. This is proof that there are still opportunities

for further enhancements of DL-based systems for DVS and showcases the importance of our work, which will assist the development of these systems.

In this Chapter, we have also presented other datasets (namely the COCO and the DeepStab datasets) and another evaluation metric that we are going to utilize in this thesis (VMAF). Both are not a result of our work but are widely used in the industry and academia, and we believe they are useful in the development of this thesis.

Chapter 4

STN-Based Image Alignment

4.1 Motivation

The DL-based DVS problem is not easy to be addressed, once camera path smoothness is hard to define in this context. In this thesis, we propose a novel video stabilization technique, which leverages Deep Learning-based image alignment and traditional camera path smoothing. In this Chapter, specifically, we present our image alignment module, which will be the starting point of our DVS proposals.

With this Chapter, our main contribution is to build the base network for this thesis, by training and testing an STN-based image alignment module. We demonstrate, with our experimental results, the good performance of our STN.

4.2 Image Alignment Module

In this Section, we present our STN-based image alignment system, depicted in Figure 4.1, which shows our workflow: we first train our DL-based image alignment. Then,

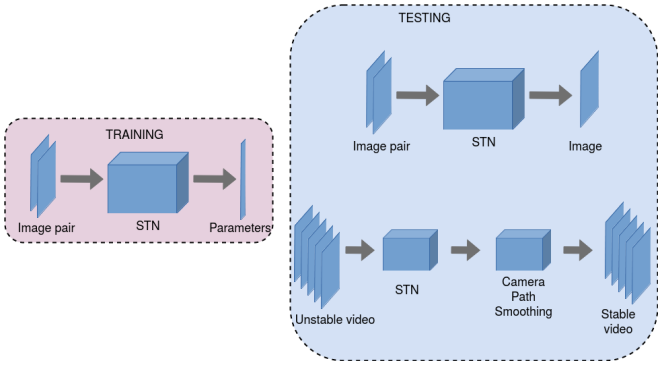


Fig. 4.1 Our system workflow.

we test it with image datasets, and after having successful results, we propose different DVS systems. An STN consists of three blocks, namely the localization network, the grid generator, and the sampler.

In the first step, our initial task is to align images, hence we do not need videos to train or test: we can use image datasets, which simplifies the training process. To this end, we use the COCO dataset [5], shown in Figure 3.3. We provide a large video dataset, which consists of hundreds of videos, each containing hundreds of frames, providing, therefore, thousands of images. However, the frames provided in each video contain basically the same scene and a moving camera, perhaps moving objects, leading to a high number of very similar images, which is not ideal for training an image alignment network, at the risk of overfitting. Hence, we use COCO as an alternative dataset, which contains natural scenes, with objects in the background, partially occluded, and amid clutter.

During training (Figure 4.2), the input to our system is a single image: we use randomly generated motion parameters to transform it and feed our network with a stack of two 160x120 images (original and transformed). The localization network then outputs an array (1x8) of motion parameters. We found out that using only the localization network of the STN is enough for training, which simplifies and speeds up the process.

In fact, we have trained the network with the entire STN and with only the localization network, and obtained similar performances. Therefore, in this case, we calculate L2 loss between the input and output motion parameters.

Although the input to our localization network is a stack of 160x120 images, the input size to our system can be of any dimensions, since prior to feeding the localization network we perform scaling and, if necessary, cropping, as seen in Figure 4.3. This does not compromise performance, since most of the feature map is used in the motion estimation, and allows for flexibility when inputting images, for either training or testing. One should note that, since the input to our system can be of any dimension, so can the output, since we transform the original image.

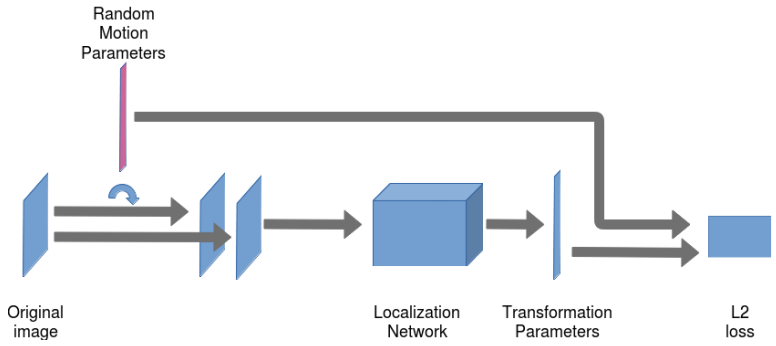


Fig. 4.2 Training procedure for the localization network.

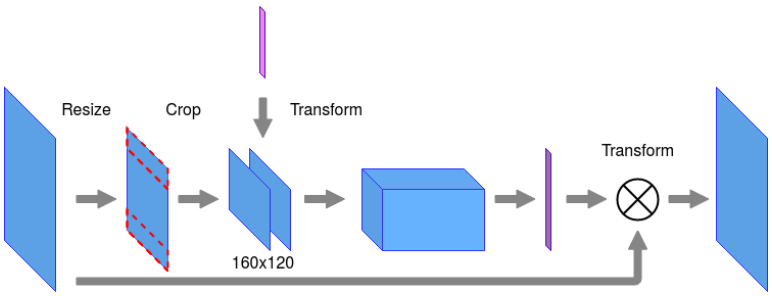


Fig. 4.3 Image dimension management performed by our system.

Our testing process is divided into two steps:

1. **Single images** (Figure 4.4): our goal is to see how well our STN can align two images. Same as in training, we feed one image, transform it with random motion

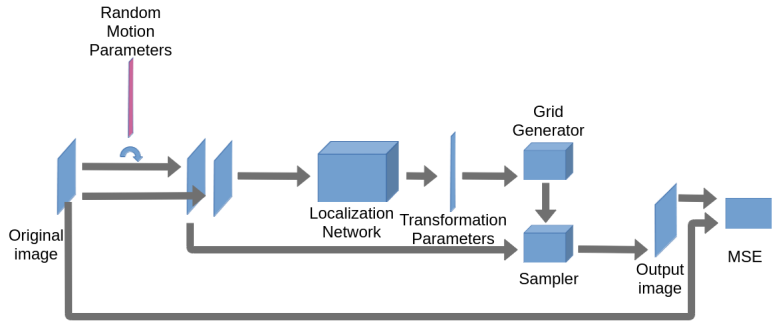


Fig. 4.4 Testing procedure for single images: image alignment.

parameters, and stack it with the original one. In this case, we utilize all the blocks in our spatial transformer, by estimating the motion parameters and then transforming the originally warped image, to obtain an output image that is as similar as possible to the original one.

2. **Videos**: our goal is to remove unwanted motion from unstable videos, without adding any other undesirable motion. To this end, we have developed a number of DVS systems. First, in this Chapter we show a simple experimental result with aligning frames of an unstable video, with simple alignment between consecutive frames. Then, we detail the implementation and behaviour of our proposed DVS systems in Chapters 5 and 6.

4.3 Experiments

4.3.1 Experimental Setup

We design and test our image alignment and DVS systems in a Linux environment, with Ubuntu 16.04, 11GB NVIDIA GEFORCE GTX 1080 Ti. Currently, we use CUDA version 9.0.176, CUDNN v7, and tensorflow version 1.7.0.

4.3.2 Experiments with COCO Dataset

We first train and test our network using the COCO dataset. This Section shows our experimental results for image alignment. We run the test with 1500 images from the COCO [5] test dataset, since we consider it to be a good sample for our purposes. In our tests, we include images that present a challenge for traditional DVS systems, such as dark scenes, and close objects. Figure 4.5 shows a sample of original image (column a), randomly transformed image (column b), and output image to the system (column c).

Fig. 4.5 Image test for the COCO Dataset. Column a: Ground truth. Column b: randomly transformed image. Column c: output.

Apart from the border effect throughout the process, the output is similar to the original image: it is possible to notice the border effect in the output image, even with the scaling factor. There is a cascade impact on the border of the output images, since they are the transformation of a previously transformed image, which would not be as severe in a regular scenario, where we would estimate the transformation parameters between two images that do not contain border effect. The metrics we study in this step are:

1) **MSE1** is calculated between the two inputs to the network, original and transformed,

2) **MSE2** is obtained between the output to the network and the original image,

3) **MSE3** is determined between the randomly selected motion parameters (PR1) and the estimated ones (PR2),

4) the mean of $PR1^2$. We use PR1 values as a baseline for comparison against MSE3, the difference between PR1 and PR2.

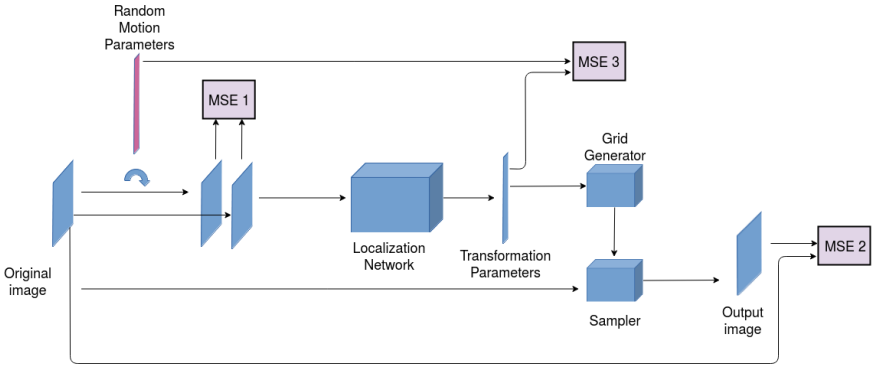


Fig. 4.6 MSE1, MSE2, and MSE3 calculation.

A diagram containing MSE1, MSE2, and MSE3 is shown in Figure 4.6, to demonstrate how we calculate these metrics. MSE3, specifically, is a measure of accuracy of our system, since it compares the ground truth (the random motion parameters) to the output to our network (the estimated parameters). Figure 4.7c is a histogram that shows

the accuracy for each of the 1500 tested images. Hence, we argue that the mean of MSE3 is a good accuracy metric: 0.00691.

Figure 4.7a shows the MSE1 values, while Figure 4.7b shows MSE2 values, both histograms have bins with a range of 0.015. MSE1 is concentrated at higher values (between 0 and 0.345), with the highest frequency between 0.045 and 0.060, while MSE2 lies between 0 and 0.18 and has the highest frequency between 0.015 and 0.030. Hence, we can conclude that the original images are considerably more similar to the output than the originally transformed ones.

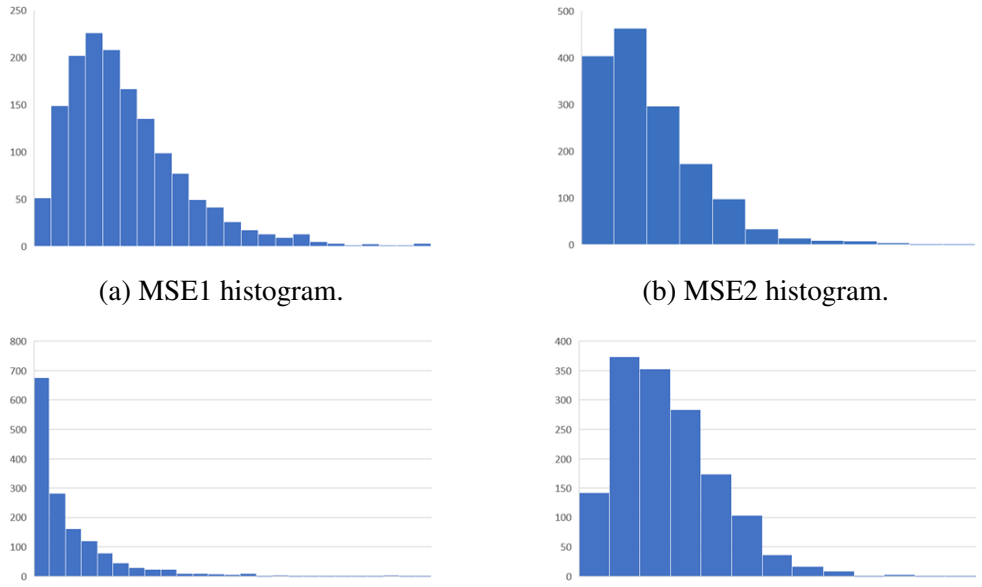
Figures 4.7c and 4.7d, which show MSE3 and mean of $PR1^2$, confirm this, both histograms have bins with a range of 0.003. MSE3 ranges from 0 to 0.075 and presents the highest frequency between 0 and 0.003. On the other hand, mean $PR1^2$ ranges from 0 to 0.036 and has the highest frequency between 0.003 and 0.006. Although the range in Figure 4.7c is wider than in Figure 4.7d, it presents nearly half of the events in the first bin, around zero, while the latter needs two bins to have approximately the same amount of events. Therefore, we conclude that the estimated motion parameters are accurately estimated by our system.

4.3.3 Experiments with our video dataset

In this Section, we carry out a performance evaluation with one video from our dataset. In this case, we try to align each frame to the previous one, without taking into account the interdependence between all the frames in the video. In other words, we try to align the frames as if they were isolated image pairs. For performance evaluation, we compare the unstable, stabilized, and GT videos, on a frame-by-frame basis.

Our metrics in this scenario are:

1) $MSE_{stabilized}$ is calculated between the GT and the stabilized frames,



(a) MSE1 histogram.

(b) MSE2 histogram.

(c) MSE3 histogram.

(d) Mean of PR1[2]

Fig. 4.7 Histograms for the image alignment experiments: COCO Dataset.

2) $MSE_{unstable}$ is obtained between the GT and the unstable frames.

In Figure 4.8, we show the histograms to $MSE_{stabilized}$ in the top row and to $MSE_{unstable}$ in the bottom row. Both histograms present bins with a range of 0.25. It is easy to see that the stabilized video presents the highest frequencies in the first half of the histogram, while the unstable presents the highest frequency in the second half. Hence, we can conclude our image alignment module works for frame alignment as well, and make sure that our image alignment module is not guided by the black borders in the images shown in column b from Figure 4.5.

4.4 Outcomes

In this Chapter, we presented the base of our DVS system, an STN-based image alignment module. Such module can take images of any dimension and output images with the same size. It utilizes an STN to estimate the motion parameter between two

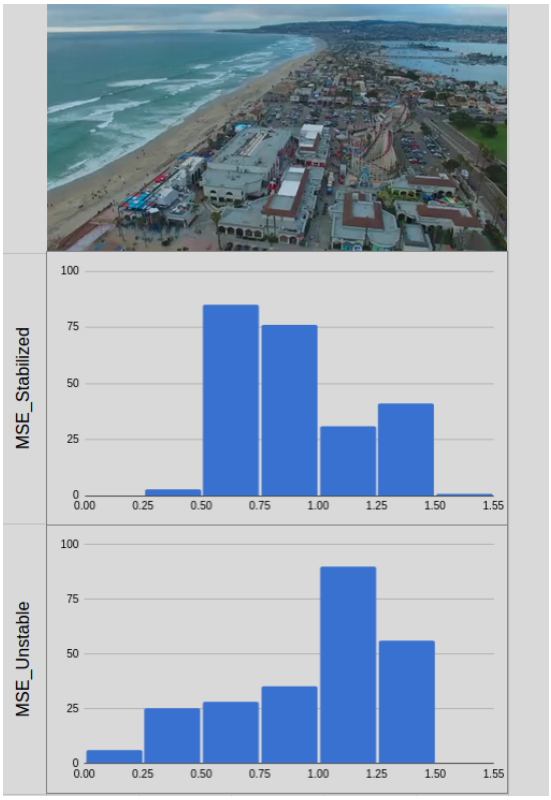


Fig. 4.8 Frame alignment test for our dataset: $MSE_{stabilized}$ (top) and $MSE_{unstable}$ (bottom).

images, then transforms one of them, to align it to the previous one. In our experiments with images, we showed that the output image is more similar to the input than the initially transformed image. We also proved that the MSE between input and output motion parameters is negligible when compared to the initial motion parameters. In our experiments with frames from videos from our dataset, we could also conclude that our image alignment module also works for aligning frames from a video, which do not contain black borders.

Chapter 5

Deep Homography-Based Video Stabilization

5.1 Motivation

Traditional and DL-based DVS methods present advantages: while the former can smooth the camera path effectively, the latter can extract and track features successfully, and estimate motion parameters between image pairs. On the other hand, both present drawbacks: the former struggles with feature detection and tracking in certain scenarios, whereas with the latter it is hard to define camera path smoothness.

This Chapter proposes a novel approach for DVS, by utilizing the benefits of DL and traditional methods. To this end, we leverage the presented image alignment module to learn motion parameters between frames. To stabilize videos, we estimate the motion parameters between the current frame and previous frames from a given unstable video, then we utilize moving average to provide long term as well as long and short term smoothness.

Depending on the jitter of the unstable video, one of these outputs will present the best results. In this Chapter, we present a solution with traditional homographies, i.e., we use all the homography parameters to perform image transformation. Because of the STN image alignment and the moving average-based camera path smoothness, we argue that our system uses the benefits of traditional and modern video stabilization.

With this Chapter, our contribution is threefold: 1) since we use STNs for aligning the frames, we avoid relying on traditional feature extraction and tracking and optical flow, which are the main reasons traditional DVS methods fail. 2) we address the problem definition issue in a DL context, by using a traditional method for camera path smoothing, which tends to be effective. 3) since we train our network with images, our proposed system does not require video datasets that are specific for training DVS systems, simplifying the process.

5.2 Deep Homography-Based DVS

In our proposed approach, we use the ability of our Image Alignment module to stabilize videos, as shown in Figure 5.1. Because we perform motion parameter estimation between two frames, for stabilizing videos we can use the same trained model utilized in Chapter 4. Unlike previous DL-based DVS methods, our proposed system does not require video pairs for training, but single images. This simplifies the training procedure, once any image dataset can be used.

Our goal is to remove unwanted motion from the unstable video, which presents high and/or low frequencies, without adding any other unwanted motion. Therefore, we feed an unstable video to the system, and similarly with the image alignment step, we perform several transformation parameter estimations. Instead of original and warped images, each transformation is estimated between different frames of a given video.

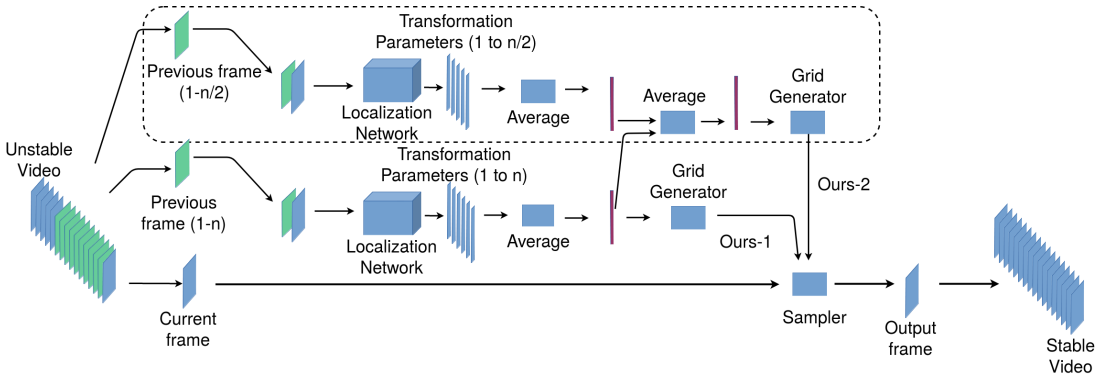


Fig. 5.1 Homography-based Video Stabilization.

If we try to align a given frame to a single previous frame, each frame transformation would be independent from the others, which would cause the output video to be wobbly, with an even higher motion frequency. Hence, considering that the current frame should be consistent with the adjacent frames in terms of motion, we compare it with a number of previous frames, for obtaining motion smoothness. For each frame, we perform the motion parameters estimation with the previous n frames. After such estimation, we provide two output videos:

i) we calculate the mean to the n motion parameter matrices and then transform the current frame, as shown in the area outside the dotted line in Figure 5.1. This moving average strategy is utilized in a number of fields for smoothing parameters, including traditional video stabilization techniques. Our current implementation considers $n = 20$ a good value, based on previous experiments.

ii) the motion parameters we utilize in this case consist of the mean between two different means: we calculate the mean to n and $n/2$ motion parameter matrices, estimated between the current frame and the previous n and $n/2$ previous frames, respectively, as shown in Figure 5.1. Then, we take the mean between them. With this output, we perform not only a long term smoothing procedure, but also a short term one,

which allows us to filter motion in different frequencies (low and high). Depending on the video motion, one of these outputs will provide better results.

In preliminary tests of the video stabilization step, we noticed that the scale parameter of the transformation matrix distorts the output videos to our system. Therefore, unlike in the image tests, for videos we limit how this parameter varies: we do not allow it to be higher than 2 times the previous value, or lower than half of it. These limitation values have been selected during our preliminary tests. Since the output frames are not fully scaled, some of them present border effect, which is not desired, since it impairs the quality of experience of viewers. Because of that, we perform a post-processing step, by cropping 10% of the borders from the width and height of each frame (5% each side, i.e. left and right, top and bottom).

Although this moving average approach has been utilized in traditional DVS methods, we have the benefit of avoiding traditional feature extraction and tracking with our approach. In fact, this is where traditional methods fail, since sometimes the video contains scenes that present little to no features to be tracked, with e.g. textureless and dark content. Hence, we argue that our system leverages the benefits of traditional and modern video stabilization, since we use STNs for motion estimation and moving average for camera path smoothing.

5.3 Experiments

In this Section, we compare our proposed system to a number of state-of-the-art proposals. We carry out two different Full-Reference performance evaluations:

1) Since we provide a dataset that is specific for DVS, with GT and unstable videos, we initially compare our system to proposed systems that are available to the public, by

either providing source code or because they are available in mobile phones. Also, in this performance comparison we use our evaluation framework.

2) In an attempt to use a widely utilized metric that considers the Human Visual System (HVS), we also use VMAF.

Fig. 5.2 Test videos [6].

Fig. 5.3 Artifacts introduced by Ours-1, StabNet, Estadeo, and GPhotos.

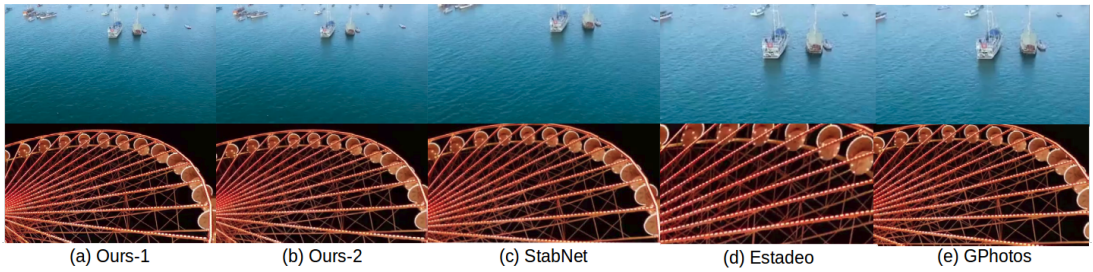


Fig. 5.4 Sample video output.

5.3.1 Performance Evaluation: our Evaluation Framework

In this Section, we carry out the performance evaluation mentioned in item **1)** and compare our system to two state-of-the-art proposals and one commercial solution. We choose these systems as baseline because they are available to the public, by either providing their source code or being present in regular consumer devices. The baseline utilized in this Section are:

1. **StabNet [23]**: a low-latency, real-time, DL-based method. It consists of a siamese network that learns a set of transformations for each input frame, considering the previously stabilized frames in the video. We use the code and pre-trained model provided by the authors.

2. **Estadeo [28]**: a paper dedicated to implementing and exhaustively comparing classic digital video stabilization techniques and boundary conditions. We utilize the default options in the code provided by the author.

3. **Google Photos [137]**: a commercial solution provided by Google, available on mobile phones. For testing this application, we upload the unstable videos to an Android phone and, in the Photos application, we select the options to edit and stabilize the videos.

We encourage the reader to refer to Chapter 2 for a thorough description of StabNet and Estadeo. Since our evaluation framework runs frame-wise, we use ffmpeg [130] to extract frames from each test video.

Our system has two outputs. In the next Figures presented in this Chapter, Ours-1 represents item *i* (long term smoothing) in the given Section, whereas Ours-2 represents item *ii* (long and short term smoothing).

For testing, we use the videos we provide in our dataset. A sample of the test videos is shown in Figure 5.2.

DVS systems should aim at obtaining not only stable videos, but also videos without blur, distortion, and that do not present high crop ratio. However, the tested systems tend to fail in one or more of the requirements for DVS, as shown in Figure 5.3. The most common artifact introduced by the tested systems is distortion, since stabilization systems normally warp the frames, which tends to distort edges. Another common artifact is crop and zoom, to remove border effect. Also, a consequence of the zoom factor is the added blur to the frames.

However, artifacts are not present in all output videos. In fact, there are several success cases. Figure 5.4 shows a sample of the output from each of the tested systems. We can note that our systems present less zoom and crop ratio, and that the blur introduced is negligible when compared to our baseline.

Although Estadeo and Google Photos (which we will name GPhotos for the sake of simplicity) output videos with the same resolution as the input, StabNet and our system output videos with different resolutions. Because of that, for testing both systems we crop every ground truth video so that they have the same dimensions.

We carry out our performance evaluation using 60 videos from our dataset and proposed evaluation framework. While we encourage the reader to refer to Chapter 3 for reference, here we mention the utilized metrics for the sake of convenience: 1) MSE, 2) SSIM, 3) Resolution Preservation, and 4) Mean distance between features.

Figure 5.5 and Table 5.1 show the average results for all measurements, the latter also presenting the standard deviation to the measurements, for the sake of completeness. Figure 5.6, on the other hand, shows the experimental results for specific videos, selected from the videos that have been tested. We show such a chart to better illustrate the performance of the evaluated systems.

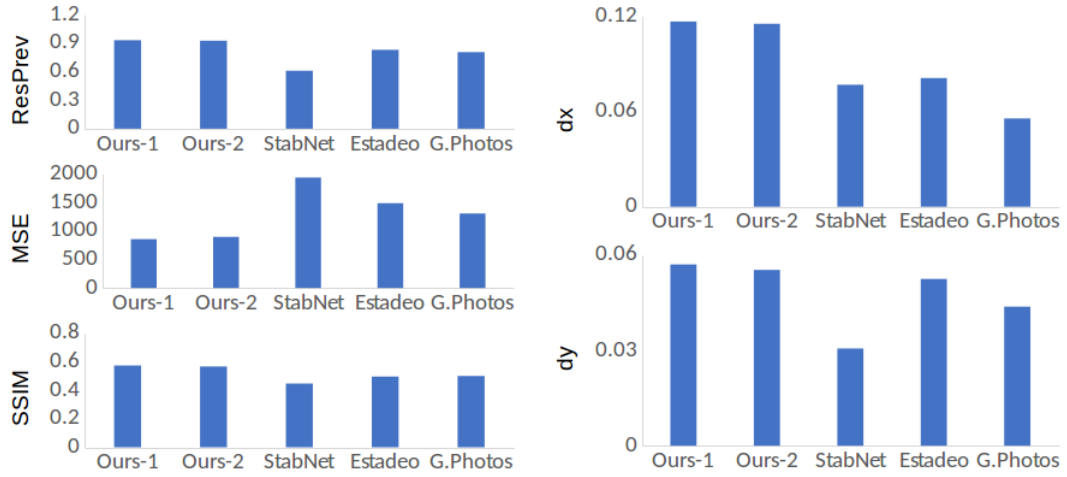


Fig. 5.5 Experimental Results for homography-based DVS (our evaluation framework): average results.

In Figures 5.5 and 5.6, we note that, overall, our system presents betters MSE results. This occurs because we limit the scale factor of our system, keeping the same image ratio (i.e., not zooming in, enlarging objects, and cropping large portions of the frame) and avoiding blur caused by scaling the frames (which is visible from the sample in Figure 5.4). The second best system in this metric is GPhotos, followed by Estadeo and StabNet. For videos 7 and 8 in Figure 5.6, all systems present high MSE: this is explained by the fact that both present challenging scenes, since video 7 contains

	RESPREV				
	Ours-1	Ours-2	StabNet	Estadeo	G.Photos
AVG	0.931	0.926	0.609	0.830	0.805
STD	0.039	0.040	0.043	0.112	0.053
	MSE				
	Ours-1	Ours-2	StabNet	Estadeo	G.Photos
AVG	855.437	896.932	1,936.189	1,487.433	1,306.550
STD	909.062	932.829	1,315.196	1,261.247	1,104.298
	SSIM				
	Ours-1	Ours-2	StabNet	Estadeo	G.Photos
AVG	0.574	0.567	0.447	0.497	0.500
STD	0.199	0.200	0.204	0.216	0.217
	dx				
	Ours-1	Ours-2	StabNet	Estadeo	G.Photos
AVG	0.117	0.115	0.077	0.081	0.056
STD	0.098	0.101	0.062	0.095	0.072
	dy				
	Ours-1	Ours-2	StabNet	Estadeo	G.Photos
AVG	0.057	0.055	0.031	0.053	0.044
STD	0.025	0.024	0.013	0.024	0.024

Table 5.1 Experimental Results for homography-based DVS (our evaluation framework).

occlusion and video 8 contains textureless content. The SSIM results show similar performance comparison, with our system presenting the best results, followed by Estadeo and GPhotos. In this case, the results for videos 1 and 4 in Figure 5.6 are the worst, which can be explained by the fact that the stabilized videos presented higher distortion, leading to higher structure difference from the ground truth, which is taken into account by SSIM calculation.

The resolution preservation (ResPrev) charts show that, although we crop 10% of each frame on both height and width, our system can still preserve more information. This is caused by the fact that we limit the scale factor in our frames, keeping edges sharp and avoiding blur.

The dx and dy charts show the normalized mean distance between features. Comparing both charts, we can note that the displacement in the x-axis is normally higher than in the y-axis for all systems. Hence, the evaluated systems provide better alignment in the

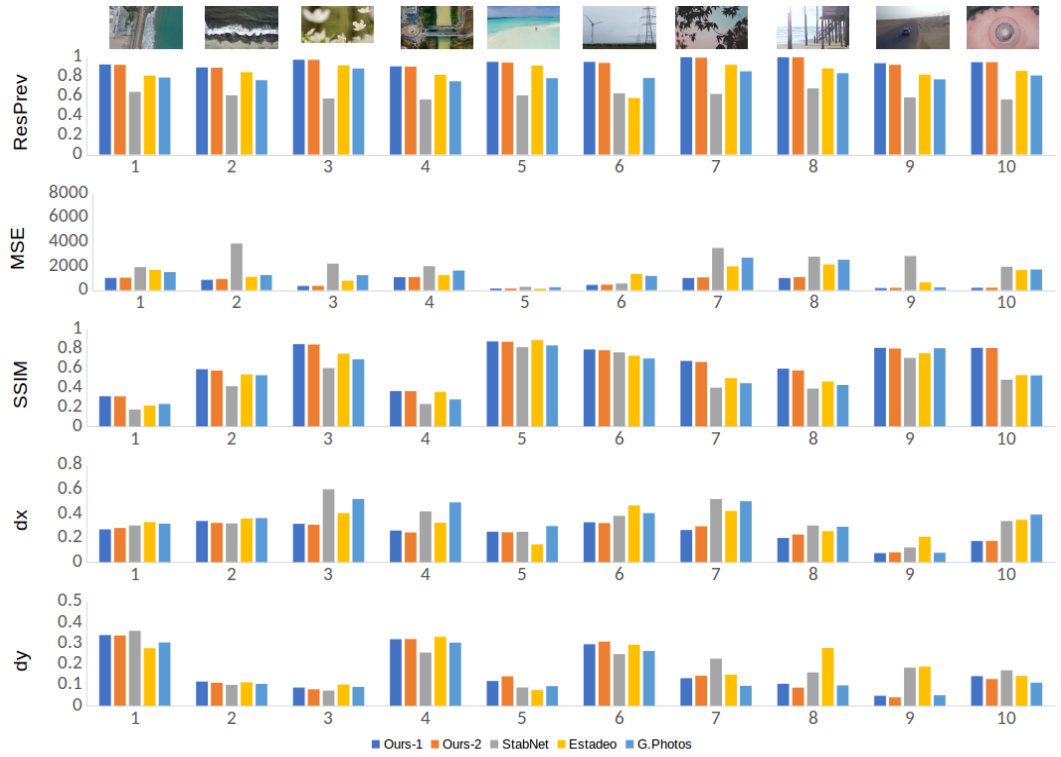


Fig. 5.6 Experimental Results for homography-based DVS (our evaluation framework): video specific results.

y-axis. We also note the difference in performance between our two systems. This is caused by the camera path in the different videos, which causes the long term (present in both systems) and short term smoothness (present in system ii) to present different performances.

In these metrics, our system does not perform as well as our baseline. This is a compromise that we make, by avoiding distorting and zooming the frames to our videos, thus allowing remaining jitter. We argue that the viewing experience of an end user and the performance of an application do not only rely on the stability of the video, but also on the blur, distortion, and scale of the stabilized video. In all these other characteristics, our system provides better performance. Therefore, we can conclude that our system outperforms the state-of-the-art proposals and the commercial solution.

Also, since we use a wide variety of content in our tests, it is possible to conclude that our proposed systems are robust enough for video stabilization.

5.3.2 Performance Evaluation: VMAF

In this Section, we carry out the performance comparison mentioned in item **2)**. Figures 5.7 and 5.8 show average and video specific results. Both charts show VMAF scores divided by the maximum VMAF score (100). In the average results, we can see that our system provide overall higher VMAF scores. On the other hand, in some scenarios our systems do not deliver the highest VMAF scores, as is the case for videos 2 and 10. Overall, although our systems do not deliver the highest VMAF scores for all videos, we argue that they are better than the state-of-the-art proposals.

Since VMAF is based on the Human Visual System, we argue that there is a subjective factor to this metric. The obtained VMAF scores only confirm the superiority in performance of our system, once VMAF has shown overall better performance of our system.

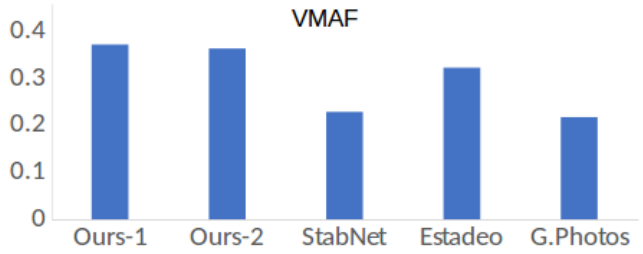


Fig. 5.7 Experimental Results for homography-based DVS (VMAF): average results.

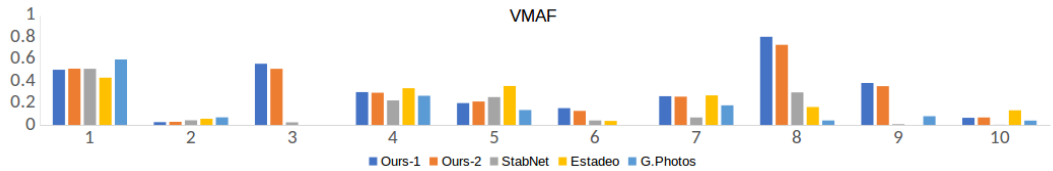


Fig. 5.8 Experimental Results for homography-based DVS (VMAF): video specific results.

5.4 Limitations

The currently proposed system performs homographies to the unstable frames to generate a stable video. However, in some cases the homographies tend to distort the output frame, generating unrealistic scenes. Figure 5.9 shows some examples, with buildings that are not straight and people walking diagonally. It also highlights the artifacts added by interpolation to the output frames. Our current implementation uses bilinear interpolation to perform homographies, which adds blur to the resulting images.

The effect of interpolation is also shown in Figure 5.10, with some other cases of blur caused by interpolation. It is easy to see that the edges of the objects lose sharpness, which is annoying, especially in high resolution videos.

Having this in mind, we next propose a system that mitigates these issues, by using translation only. Instead of performing homographies, we simply move the frames in the x and y axis, not manipulating the content of the frame. Hence, it allows us to preserve the sharpness of the images, and to avoid distotion. Although the translation only video stabilization allows some jitter left in the video, we argue that the preservation of crucial features of the frame is as important as stabilizing a video for the viewing experience.

5.5 Outcomes

In this Chapter, we proposed a video stabilization system based on traditional and Deep Learning methods by combining the advantages of STNs and their ability to align two images, and the simplicity of smoothing videos with moving averages. Our system is a simple approach that is more robust to scene content than traditional methods. With our moving average approach, we addressed the path smoothing definition issue

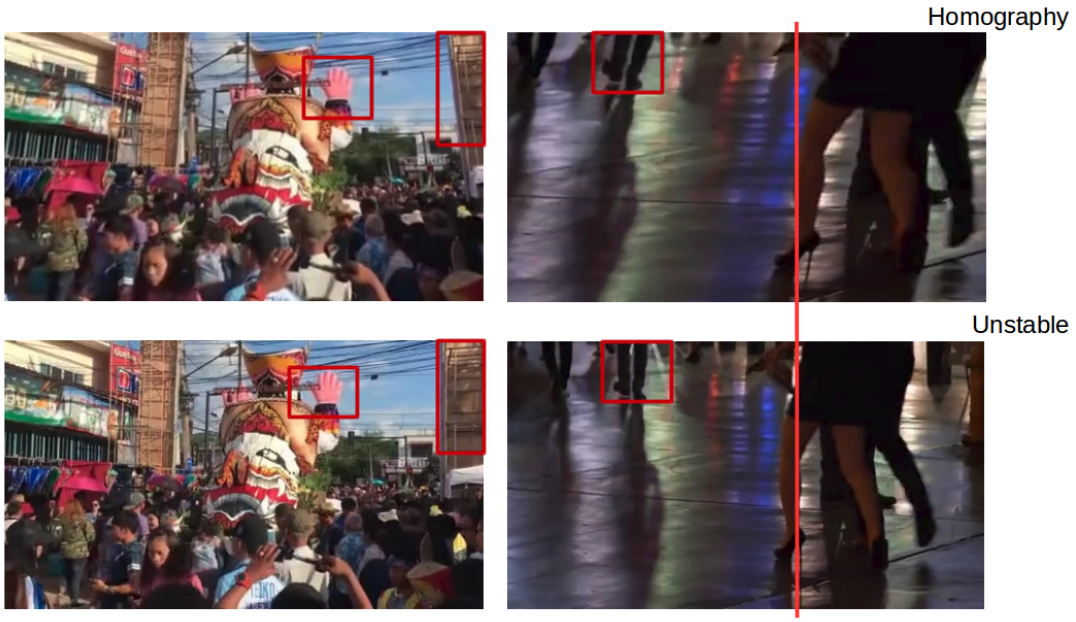


Fig. 5.9 Artifacts added by homography: distortion and blur.

video stabilization systems face in a DL context. To stabilize the input videos, we use homographies to transform each frame. Experimental results showed that our system can stabilize videos even in challenging scenarios, in which traditional methods tend to fail. They also showed that our system outperformed the state-of-the art proposals and commercial application with our evaluation framework and the VMAF score.

Although the system we proposed so far presents good performance, it is not ideal. Since we perform homographies to every frame of the unstable video, we end up interpolating them. Regardless of the interpolation we perform, we end up adding blur to the frame, which is undesirable. Also, since we perform translation, rotation, and scaling (to some extension), we may add distortion to the output frames. These artifacts are annoying to the viewer, and could impair the performance of any computer vision application.

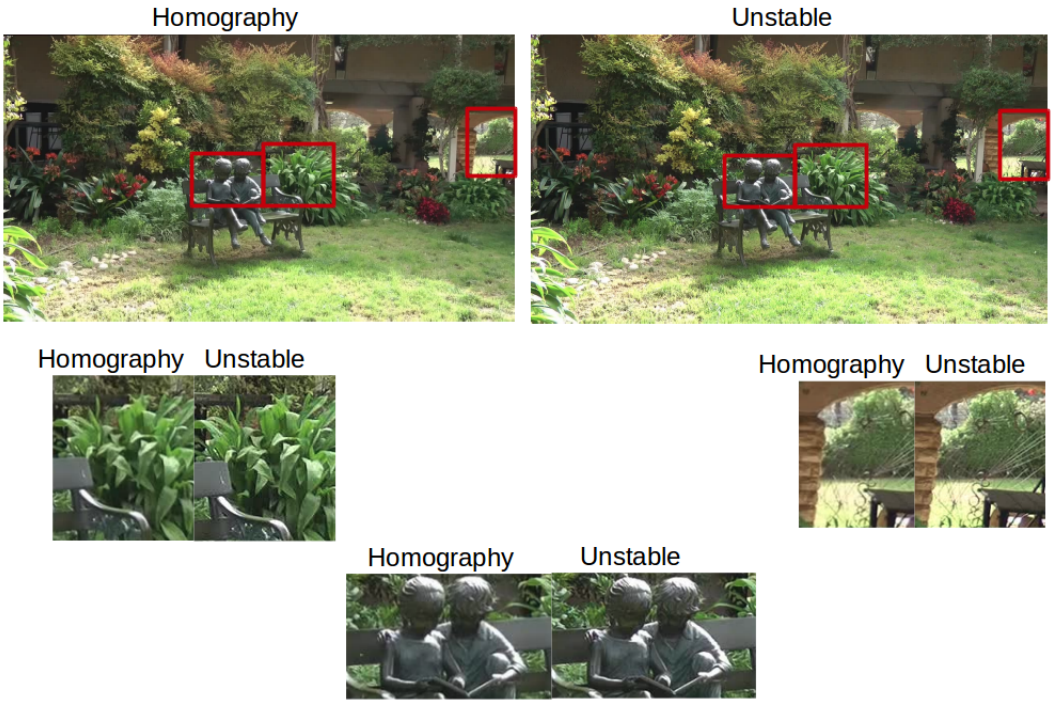


Fig. 5.10 Artifacts added by homography: blur.

Chapter 6

Deep Translation-Based Video Stabilization

6.1 Motivation

Two common artifacts added to stabilized videos by DVS systems are blur and distortion. These tend to annoy the viewer, or to damage the performance of an application that uses the stabilized video, and should be avoided in the stabilization process.

In fact, most of the DVS solutions available in the academia do not address the blur factor in the original video. Added to that, since most DVS systems perform 2D transformations to the frames, they tend to distort the stabilized frames and cause additional blur, which are the most noticeable artifacts in stabilized videos [14].

In this Chapter, we continue with our approach to video stabilization, by leveraging STNs to learn motion parameters between two consecutive frames. Then, unlike our homography-based approach, we use Exponentially Weighted Moving Average (EWMA), a traditional technique, to smooth the camera path. We also do not perform

homographies, hence avoiding 1) distortion and 2) interpolation, which tend to blur images.

Instead, we estimate how much a frame should be translated in the x and y axis and pad the edges with zeros, then crop the frame to avoid border effects. Since our camera path smoothing only performs translations in the x and y axis, we allow certain level of jitter left in the ouput video, in a tradeoff between video stability and overall frame quality. We believe the outcomes of this Chapter are essential for improving the quality of amateur videos, hence improving the viewing experience of such content and allowing good performance of computer vision applications that rely on the stabilized videos.

With this Chapter, our main contribution is to avoid performing homographies on the video frames, avoiding blurring and distorting the resulting frames by simulating translation in the x and y axis.

6.2 Deep Translation-Based DVS

Our approach follows the main idea presented in our homography-based DVS, which aims at removing unwanted motion from the unstable video without adding any other unwanted motion. We feed an unstable video to the system, and perform several transformation parameters estimations. Instead of original and warped images, each transformation is estimated between different frames of a given video. Having in mind that the current frame should be consistent with the adjacent frames in terms of motion, we compare it with a number of previous frames, for obtaining motion smoothness. For each frame, such estimation is performed n times, between the given frame and the previous n frames. Then, we calculate the EWMA of the given values and take two different averages:

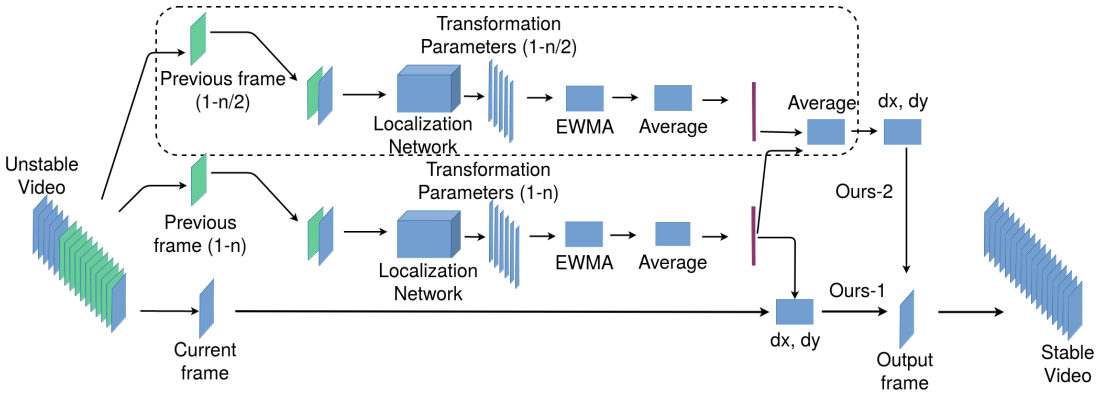


Fig. 6.1 Translation-based Video Stabilization method.

i) we calculate the mean to the n motion parameter matrices, as shown in the area outside the dotted lines in Figure 6.1. With this output, we perform long term smoothing.

ii) as shown in Figure 6.1, the motion parameters we utilize in this case consist of the mean between two different means: the EWMA to the n motion parameters and the EWMA to the $n/2$ motion parameters. With this output, we perform long and short term smoothing, which allows us to filter motion in different frequencies (low and high).

Depending on the video camera path, one of these outputs will provide better results. Our current implementation considers $n = 20$ a good value, based on preliminary experiments. After obtaining the motion parameters, we simulate image translation in the x and y axis, according to Figure 6.2. We estimate how many pixels the frame would move with a regular image transformation (x and y translation only). Then, we pad the frame, by adding the estimated amount of rows and columns with zeros at the edges of the frame, and cropping the opposite side. Since we do not perform any image transformations (i.e., we do not multiply transformation matrices and images), we do not interpolate the images, not adding blur to the resulting frame. Because we only perform translation, we avoid the inherent frame distortion of a traditional transformation, normally caused by rotation.

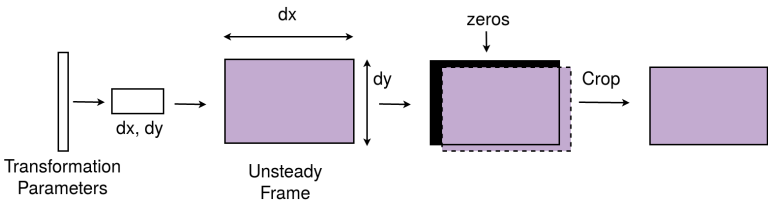


Fig. 6.2 Translation in the x and y axis.

Fig. 6.3 Test videos.

Finally, to avoid the border effect (i.e., the resulting black borders from translation), we perform a post-processing step, by cropping 10% of the borders from the width and height of each frame (5% each side: left and right, top and bottom). This would result in slightly smaller frames, which does not impair frame quality, as we show in our experimental results.

Although EWMAs have been utilized in traditional DVS methods, with our approach we benefit from not relying on traditional motion estimation, where traditional methods fail, since traditional feature extraction is sensitive to scene contents. Also, the EWMA utilization simplifies the camera path smoothing problem, which is hard to define by DL-based systems. Hence, we argue that our system leverages the benefits of both traditional and modern video stabilization, since we use STNs for motion estimation and EWMAs for path smoothing.

6.3 Experimental Results

In this Section, we carry out a performance evaluation of our translation-based video stabilization. We evaluate our systems and our baseline with a wide range of scene contents and video categories, in an attempt to observe how robust our proposed system is. For testing, we use the videos we provide in our dataset. A sample of the test videos is shown in Figure 6.3.

We run experiments with our evaluation framework. Here, we present our experiments in two steps: **1)** baseline, where we compare our system to the same state-of-the-art proposals and commercial solution from Chapter 5: StabNet, Estadeo, and GPhotos. While Estadeo and GPhotos output videos with the same resolution as the input, StabNet and our system output videos with different resolutions. Hence, for performance evaluation, we crop every ground truth video for testing both systems, so they have the same dimensions. **2)** ablation studies, where we test different settings of our system using one metric from our evaluation framework, namely Resolution Preservation.

Then, we compare the performances to our systems and baseline using the VMAF score.

6.3.1 Our Evaluation Framework

In the **baseline** step, we show a sample of the output from each of the tested systems in Figure 6.4. We can note that our system presents the output with less zoom, highest detail level, and no distortion or blur. One proof of less zoom can be seen with the red circles in the Figure: while our outputs contain the scene within the circle, the other systems either have cropped it entirely or have maintained it, close to the border. It is proof that our efforts in not performing image transformation on the video frames work. However, since our camera path smoothing only performs translations in the x

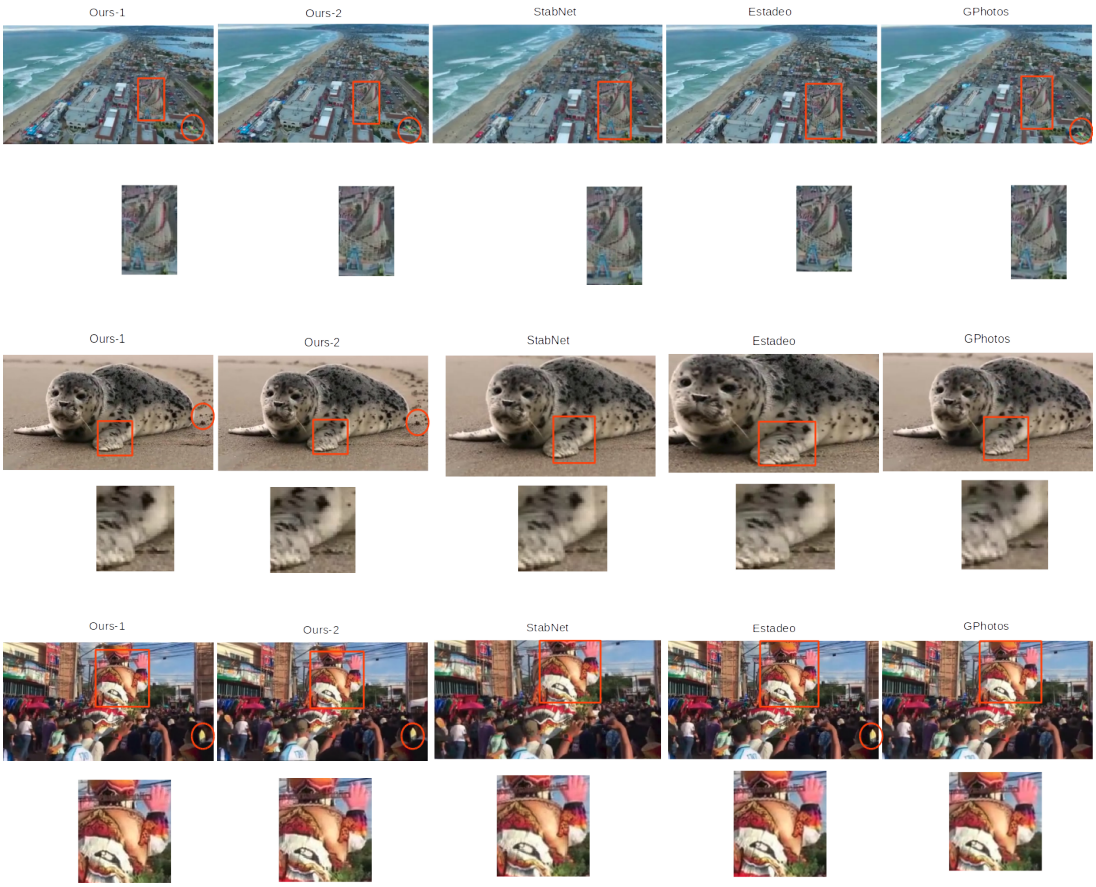


Fig. 6.4 Sample outputs for the translation-based DVS.

and y axis, we allow certain level of jitter left, in a trade-off between video stability and overall frame quality.

Since the evaluation framework we utilize [6] runs frame-wise, we use ffmpeg [130] to extract frames from each test video. Figures 6.5 and 6.6 show the experimental results for the video tests, the former showing average results and the latter, video specific results. Table 6.1 shows a more thorough experimental result, with the addition of the standard deviation, for the sake of completeness. For plotting the chart in Figure 6.6, we select a group of 10 videos from the dataset provided by [6]. Most videos present a challenge, and are not easily stabilized by traditional methods: they are either dark, present high motion, parallax, zoom, moving camera, crowds, textureless objects, or

	RESPREV				
	Ours1	Ours2	StabNet	Estadeo	G.Photos
AVG	0.978	0.977	0.609	0.830	0.805
STD	0.023	0.023	0.043	0.112	0.053
	MSE				
	Ours1	Ours2	StabNet	Estadeo	G.Photos
AVG	744.489	770.095	1,936.189	1,487.433	1,306.550
STD	783.345	809.539	1,315.196	1,261.247	1,104.298
	SSIM				
	Ours1	Ours2	StabNet	Estadeo	G.Photos
AVG	0.610	0.607	0.447	0.497	0.500
STD	0.192	0.193	0.204	0.216	0.217
	dx				
	Ours1	Ours2	StabNet	Estadeo	G.Photos
AVG	0.096	0.102	0.077	0.081	0.056
STD	0.071	0.075	0.062	0.095	0.072
	dy				
	Ours1	Ours2	StabNet	Estadeo	G.Photos
AVG	0.056	0.056	0.031	0.053	0.044
STD	0.025	0.024	0.013	0.024	0.024

Table 6.1 Experimental Results for translation-based DVS (our evaluation framework).

occlusion. We utilize a wide range of scene contents and video categories, in an attempt to observe how robust our proposed system is. As mentioned in Section 6.2, our system has two outputs. In these Figures, Ours1 represents the long term smoothing output (*i*), whereas Ours2 represents the long and short term smoothing output (*ii*).

Overall, our system presents better MSE results. This is expected, since we do not transform the frames of the input videos, avoiding distortion and blur. The second best system in this metric is GPhotos, which presents similar MSE values to ours in some cases. For videos 2, 7, and 9, all systems present high MSE: they present challenging scenes, since they contain high motion, close object, and zoom. For SSIM values, our system presents the best results, followed by GPhotos and Estadeo.

The ResPrev results show that, although we crop 10% of each frame on height and width, our system can preserve more information. This is another proof that the lack of image transformation allows frame quality preservation and its sharpness. The second best system in this metric is Estadeo, followed by GPhotos.

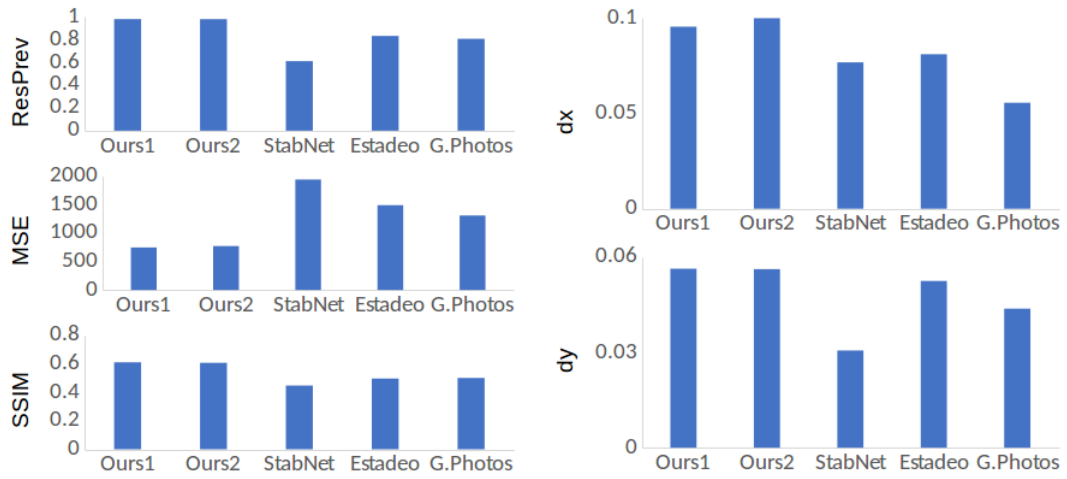


Fig. 6.5 Experimental Results for translation-based DVS (our evaluation framework): average results.

The dx and dy charts show the normalized mean distance between features in the x and y axis. Comparing the scale in dx and dy, we can note that the distance in the x axis is normally higher than in the y axis for all systems (which agrees with our results in Chapter 5).

We also note a slight difference in performance between our two systems: this is caused by the camera path in the different videos, which leads the long term (present in Ours1) and long and short term smoothness (present in Ours2) to present different performances. In these metrics, our system does not perform as well as our baseline. In fact, this is a compromise that we make, by allowing certain jitter to remain in our resulting videos, so that we can avoid distorting and blurring the videos. On the other hand, our system provides better performance with respect to the other metrics, which are as important for the viewing experience. Hence, we conclude that our system outperforms our baseline, although there is room for improvement.

In the **ablation studies** step shown in Figure 6.6, we compare the performance of six different outputs to our system. For each of the videos tested in the baseline studies, we test: i) the output using full homography transformations (Ours1 and Ours2); ii) the

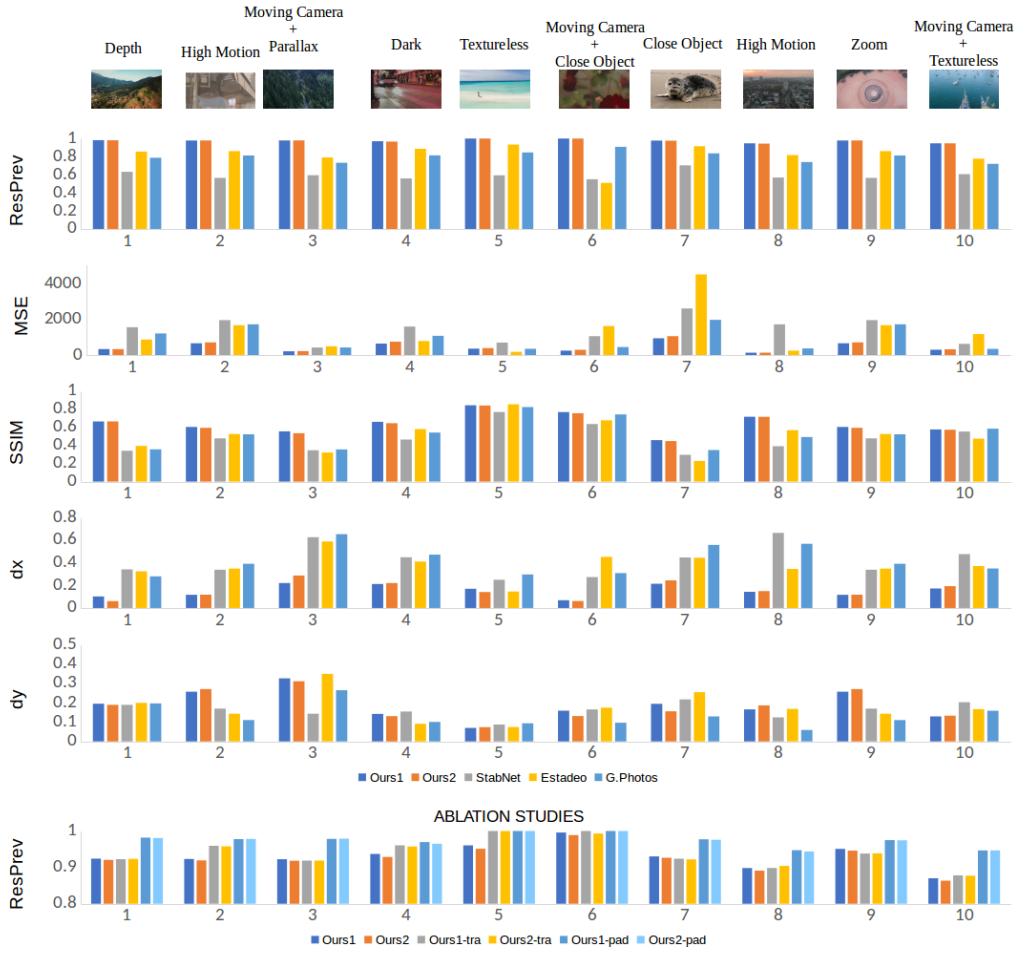


Fig. 6.6 Experimental Results for translation-based DVS (our evaluation framework): video specific results.

output using only translations in the frame transformations (Ours1-tra and Ours2-tra), i.e., we perform homographies with all but the x and y translation parameters equal to zero; iii) the output described in Figure 6.2 (Ours1-pad and Ours2-pad). Since we intend to observe how much frame quality is preserved using our method, we only show the frame preservation results.

Although the frame preservation values for Ours1-tra and Ours2-tra present good results in some cases, they are not as good as the result for Ours1-pad and Ours2-pad. This confirms our idea of not performing homographies using just the translation parameters,

since homography transformations require interpolation, even if only the translation parameters are utilized.

6.3.2 VMAF

In this Section, we evaluate the performance of our systems and our baseline using a widely utilized metric in the academia and industry, VMAF. Figures 6.7 and 6.8 show average and video specific results, respectively. The charts show VMAF scores divided by the maximum VMAF score (100).

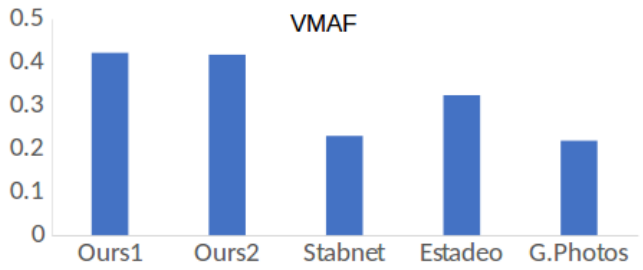


Fig. 6.7 Experimental Results for translation-based DVS (VMAF): average results.

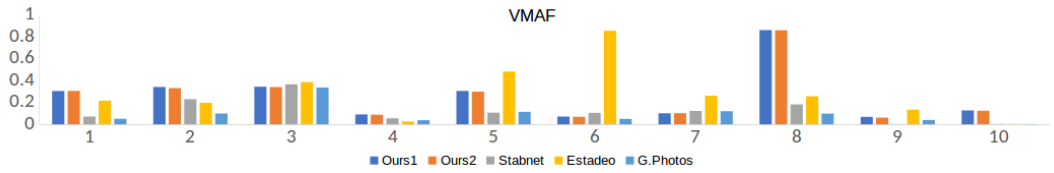


Fig. 6.8 Experimental Results for translation-based DVS (VMAF): video specific results.

The charts show higher VMAF scores for both our systems in the average results, which confirms the results in the previous Section, by showing that our proposed DVS methods are better than our baseline. However, as it is seen in Figure 6.8, for some specific videos our baseline is better, e.g. in videos 3, 5, and 6. It confirms that our systems are sensitive to content, and that we still have room for improvement. However, since these were isolated results and in average our systems provide better VMAF, we conclude that, overall, our systems outperform the state-of-the-art.

6.4 Outcomes

We proposed a video stabilization system that does not perform image transformation, to avoid blurring and distorting the resulting frames. Instead, we simulate translations in the x and y axis by padding the edges of the frames. Although this approaches allow some jitter in the stabilized video, it avoids blurring and distorting the output video, so we argue that this is a good tradeoff.

Our experimental results show that both translation-based system outperforms the state-of-the-art proposals and one commercial application. Our ablation studies demonstrate the advantages of not transforming frames to the final output.

Chapter 7

Deep Filter-Based Video Stabilization

7.1 Motivation

After the translation-based stabilization, we progress with another DVS approach. This time, we utilize a number of filters (including Kalman and low-pass), some of them have been constantly utilized for video stabilization. However, we would like to discuss their performances more deeply, by comparing them to other filters, that have not been utilized for video stabilization. We test a number of filters, and some parameters, to select one, or a few, that present the best overall performance.

We select the ideal filters for both the homography and the translation-based systems. Then, we compare their performances to our state-of-the-art. Followed by that, we compare the performances of our three proposed systems: homography-based, translation-based, and filter-based. Finally, we carry out a performance analysis that considers the possibility of using our filters added to one of our baselines, StabNet, and compare three combinations: our system, StabNet, and our system added to StabNet.

The contribution with this Chapter is threefold: 1) we carry out an in-depth filter selection for DVS systems, considering both our homography-based and translation-based systems. 2) we compare the performance of our three different proposals, selecting the best between them. 3) we consider the possibility of adding our filters to StabNet, and carry out a performance evaluation.

7.2 Deep Filter-Based DVS

So far, we have presented DVS strategies that leverage moving averages and EWMAs for smoothing the camera path of unstable videos. However, although using damping techniques for video stabilization (i.e., MVI) is a good strategy, using filters (i.e., FPS) tend to present a better performance. In this Section, we discuss our utilization of filters and their selection.

To this end, we implement a number of different filters (low-pass and Kalman), with a different set of parameters each, to remove high frequencies from the camera path. For the homography-based system, we filter each of the parameters of a homography matrix, and for the translation-based system, we filter the x and y translation. The filters we implement are: Butterworth filter, Gaussian filter, Kalman filter, and FFT filter. The Butterworth, Gaussian, and Kalman filters have been discussed in Chapter 2.

We have not previously discussed the FFT filter because of its simplicity. It consists of a frequency analysis that is performed on the camera path of the unstable video. We eliminate the highest frequencies of the camera path, considering a cutoff frequency, and keep low frequencies in a step-like process, without any smooth transition. Although this and the Butterworth filter are not widely utilized in DVS, we choose to compare them as alternatives to the popular Gaussian and Kalman filters, in an attempt to test a wide variety of possibilities for filters.

Filter	Parameter	Levels
Butterworth	Cut-off frequency	20, 30, 50, 60, 70
Gaussian	Sigma	6, 8, 9, 10, 12
Kalman	Prediction Error Covariance	1, 5, 10, 30, 50
FFT	Cut-off frequency	40, 50, 75, 85, 100

Table 7.1 Filters, Metrics and Levels utilized in our filter selection experiments.

We first obtain the accumulated global motion vector of the unstable video using our image alignment module. Then, we perform camera path smoothing using all the filters and factors shown in Table 7.1, which have been selected during preliminary experiments. Followed by that, we evaluate the performance of each filter and parameter using a stability metric *Stab*, which takes into account the *Mean distance between features*, from our evaluation framework: it measures how much a given feature has moved from the frames in the ground truth to the frames in the stabilized videos. This metric indirectly measures the stability of the output video: the less the features move from ground truth to stabilized videos, the more stable the latter is.

We use a single metric from our evaluation framework in this case because we have previously observed that our system presents similar performances for MSE, SSIM, and Resolution Preservation for different filtering methods.

Stab also considers a metric proposed by [50], which takes into account the frequency domain of the camera path. We obtain the accumulated global motion in the parameter to be filtered: a parameter from the homography matrix (for the homography-based system) or x and y translation (for the translation-based system). Then, we compute the ratio of the sum of lowest (2nd to 6th) frequency energies and the total energy, and take the minimum value. Hence, we describe *Stab* with Eq. 7.1:

$$Stab = D_f(x) + D_f(y) + \alpha \times \frac{1}{En_{low}} \tag{7.1}$$

, where $D_f(x)$ and $D_f(y)$ represent the mean distance between features in the x and y axis, En_{low} is the percentage of energy within the lowest frequencies, and α is a proportionality index, which leads all variables to be within similar lengths. With *Stab*, we select the most stable output videos from our experiments with filters and their different parameters.

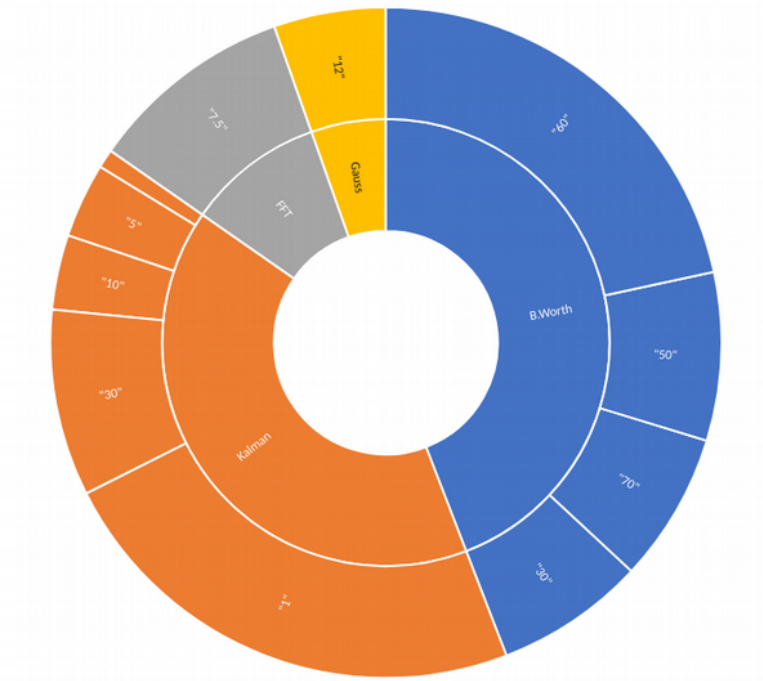


Fig. 7.1 Filters, their levels, and the amount of results: homography-based DVS.

We run our experiments on the datasets provided by [6] and [23], in an attempt to observe the largest possible number of video types and camera paths, so we can more accurately select the most suitable filter for stabilizing videos. Both datasets present a wide variety of scenarios and camera paths, which allow us to confidently choose the most robust filter out of the tested ones.

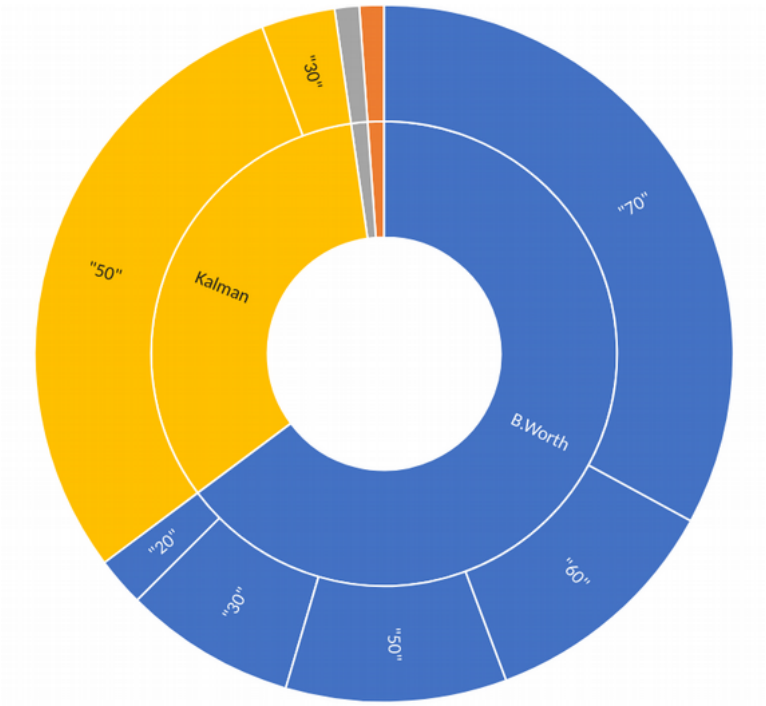


Fig. 7.2 Filters, their levels, and the amount of results: translation-based DVS.

For each video, we run our video stabilization code, using the filters and levels shown in Table 7.1. We then use our *Stab* metric to choose a range of filters that present better performance for the given video. After choosing a set of 1 to 3 best filter-parameter for each video, we obtain the sunburst charts shown in Figures 7.1 and 7.2 for the homography and translation-based systems, respectively. The inner circle of the charts show the most frequent filters, and the outer circles show the most frequent parameters.

The biggest arcs in the circles, which represent the most frequent filter and parameter, are the **Butterworth 60** and **Kalman 1** for the homography-based system and the **Butterworth 70** and **Kalman 50** for the translation-based system. It shows that, although the Kalman filter is widely utilized, it is not the best one from our test results. It is caused by the fact that such filter assumes simple environment conditions for fast calculation, which are not always the case for the camera paths. Also, as mentioned by [138], using Kalman filter with fixed parameters during the whole video is not suitable, once the camera path jitter may vary along the video. While we use the same parameter

with the Butterworth filter, is is widely utilized in gait analysis, hence we argue that its behavior is more robust to changes in the camera path, and that the the filter selection has played an important role in the effectiveness of the filter.

The other filters (i.e., Gaussian, FFT) have been selected very few times for both systems. This is a surprising result for the Gaussian filter, which has been widely utilized in the academia for DVS applications. The same argument by [138] made regarding the Kalman filter applies for the Gaussian filter: the Gaussian filter is not suitable to smooth the camera path with fixed parameters. In fact, the authors of [138] propose an adaptive Gaussian filter for smoothing the camera path. This is proof that a filter should not be selected only because of its simplicity, but its performance should be taken into account as well. As for the FFT filter, its low effectiveness is expected, once cutting off high frequencies without the proper use of a low-pass filter tends to add unwanted jitter to the original camera path.

It is clear that the Butterworth and Kalman are the most frequent filters. In the end, we select the filters and parameters shown in Tables 7.2 and 7.3 for the homography and translation-based systems, respectively.

Filter	Parameter	Levels
Butterworth	Cut-off frequency	60
Kalman	Prediction Error Covariance	1

Table 7.2 Filters, Metrics and Levels selected in our experiments: homography-based DVS.

With the selected filters, we present our new DVS system, based on filters (Kalman and low-pass), in Figure 7.3. We input an unstable video with n frames, then use our STN to obtain the accumulated global motion vector (considering both our homography and

Filter	Parameter	Levels
Butterworth	Cut-off frequency	70
Kalman	Prediction Error Covariance	50

Table 7.3 Filters, Metrics and Levels selected in our experiments: translation-based DVS.

translation-based systems separately). After that, we calculate the corrected vector using our filters for frames 2 to *n* and perform translations to frames 2 to *n*, providing a stabilized video.

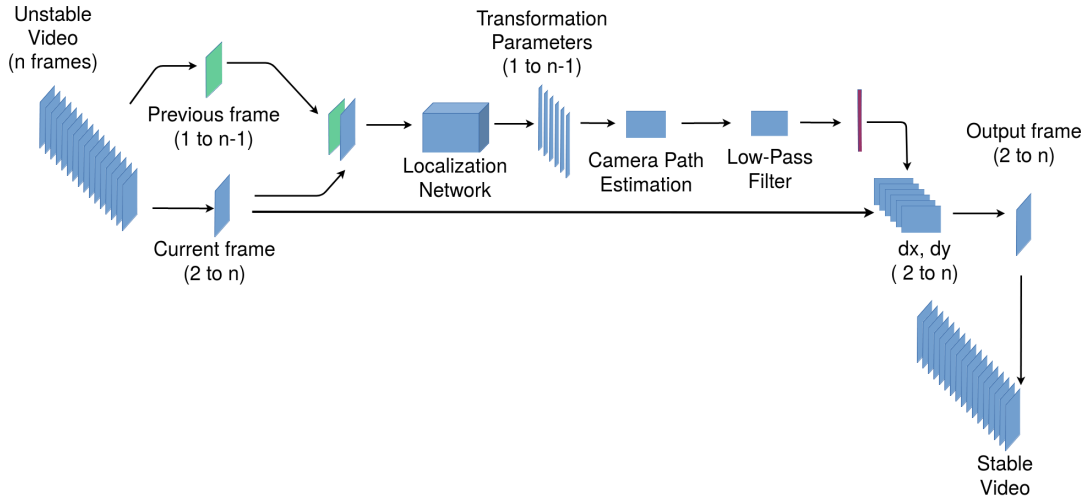


Fig. 7.3 Filter-Based Video Stabilization method.

7.3 Experimental Results

7.3.1 Our Evaluation Framework

In the **baseline** test, we first extract the frames from each video using ffmpeg [130]. In this case, we use two different filters for each system, namely Butterworth and Kalman, which we name Tra.B and Tra.K for the translation-based systems and Homo.B and

Homo.K for the homography-based systems in the charts, for the sake of simplicity. For testing, we use the videos we provide in our dataset.

Figures 7.4 and 7.5 show the average and video specific experimental results. Table 7.4 show average results, with the addition of standard deviation. For plotting the chart in the video specific Figures, we select 10 videos from our test dataset. This set of vides contains challenging and diverse content, to test how robust our systems are.

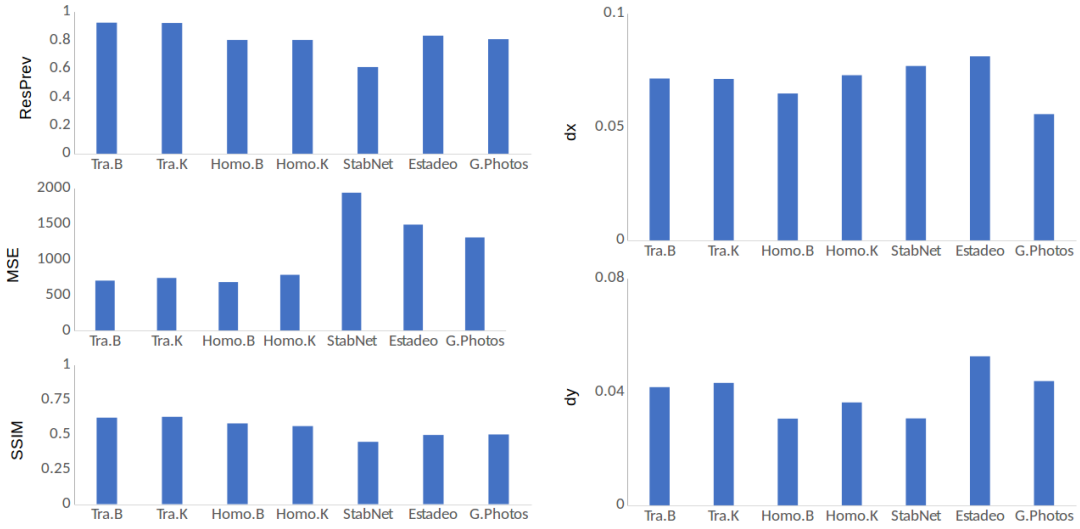


Fig. 7.4 Experimental Results for filter-based DVS (our evaluation framework): average results.

The Resolution Preservation values show that our systems present slightly better results, even though we crop the frames of our output videos, which confirms our choice of using translation only. The translation-based systems present better results than the homography-based ones. The MSE values are significantly better for the translation and homography-based filters, which shows that their outputs present higher similarity with the ground truth. SSIM results are better for our systems, although the difference in performance is not as high as the MSE values.

The dx and dy results (which show the normalized mean distance between features in the x and y axis from stabilized to ground truth videos) confirm our observations from

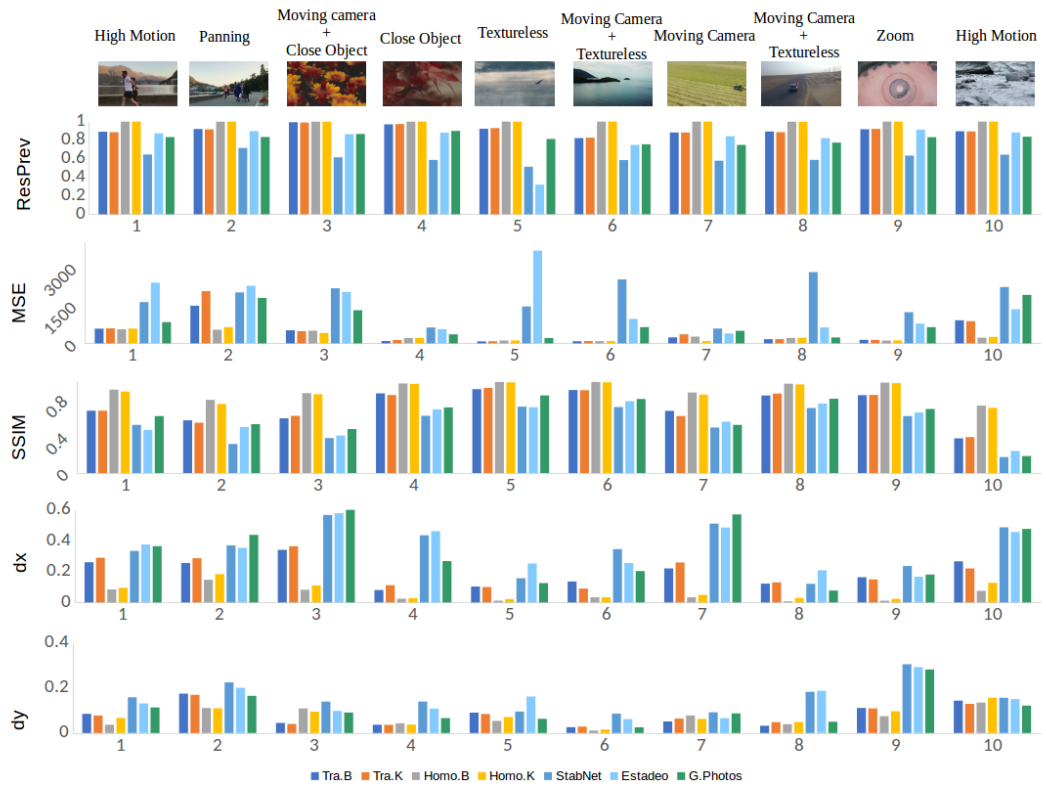


Fig. 7.5 Experimental Results for filter-based DVS (our evaluation framework): video specific results.

the previous experiments, that the x axis is not as stable in the output frames as the y axis. In this scenario, our systems present comparable results to our baseline, although they are not the best in both scenarios. In the video specific results, it is possible to observe that although our average dx and dy are not the best between the tested systems, they are still significantly better than our baseline in some cases.

The viewing experience of an end user does not only rely on the stability of the video, but also on the blur, distortion, and scale of the same video. Since our systems present overall better results in these other characteristics, we conclude that our system provides better performance than our baseline.

Fig. 7.6 Experimental Results for filter-based DVS (our evaluation framework): ablation studies.

In the **ablation studies** step shown in Figure 7.6, we compare the performance of all the filters that we have compared in our filter selection, namely: Butterworth, Gaussian, Kalman, and FFT. For most of the filters, we test a single parameter, however for the Butterworth and Kalman ones we test more than one parameter. The parameter values are presented in the legend of the chart, next to the filter name. We test eight combinations of filters and their parameters, for the feature displacement metric. We test only one metric since we have noticed in previous experiments that the performance of our systems present similar performance in the other metrics.

The latter two filters (B.Worth 60, Kalman 1 for the homography-based system, B.Worth 70 and Kalman 50 for the translation-based system) are the ones that have been selected for the performance evaluation just presented. It is clear that, although we have selected these filters, the other filter options present similar performances in some cases. However, overall the selected filters provide better performance, which confirms our selection.

	RESPREV						
	Tra.B	Tra.K	Homo.B	Homo.K	StabNet	Estadeo	G.Photos
AVG	0.922	0.920	0.800	0.800	0.609	0.830	0.805
STD	0.042	0.043	0.010	0.010	0.043	0.112	0.053
MSE							
	Tra.B	Tra.K	Homo.B	Homo.K	StabNet	Estadeo	G.Photos
AVG	699.686	736.992	678.594	781.594	1936.189	1487.433	1306.550
STD	729.733	772.923	408.040	454.904	1315.196	1261.247	1104.298
SSIM							
	Tra.B	Tra.K	Homo.B	Homo.K	StabNet	Estadeo	G.Photos
AVG	0.621	0.627	0.580	0.560	0.447	0.497	0.500
STD	0.191	0.185	0.116	0.122	0.204	0.216	0.217
dx							
	Tra.B	Tra.K	Homo.B	Homo.K	StabNet	Estadeo	G.Photos
AVG	0.071	0.071	0.065	0.073	0.077	0.081	0.056
STD	0.040	0.040	0.056	0.052	0.062	0.095	0.072
dy							
	Tra.B	Tra.K	Homo.B	Homo.K	StabNet	Estadeo	G.Photos
AVG	0.042	0.043	0.031	0.036	0.031	0.053	0.044
STD	0.051	0.052	0.044	0.045	0.013	0.024	0.024

Table 7.4 Experimental Results for filter-based DVS (our evaluation framework).

7.3.2 VMAF

In this Section, we present a performance evaluation using VMAF in Figures 7.7 and 7.8, which show average and video specific results, respectively. Both charts show the obtained VMAF scores divided by the maximum VMAF score, 100. The average results show higher VMAF scores for the Butterworth and Kalman filters applied to the homography-based and translation-based cases.

In the video specific results, our systems either present better performance or similar, which allows us to confirm that we have outperformed the baseline in both performance evaluations.

7.4 DVS Proposals Comparison

We have proposed three different approaches to DVS, namely: 1) homography-based, 2) translation-based, and 3) filter-based. In this Section, we compare the three approaches

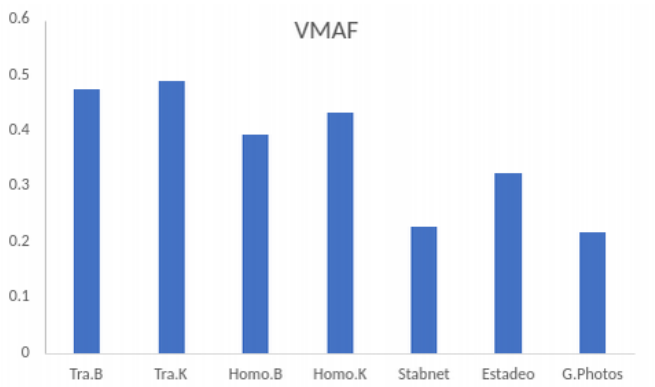


Fig. 7.7 Experimental Results for filter-based DVS (VMAF): average results.

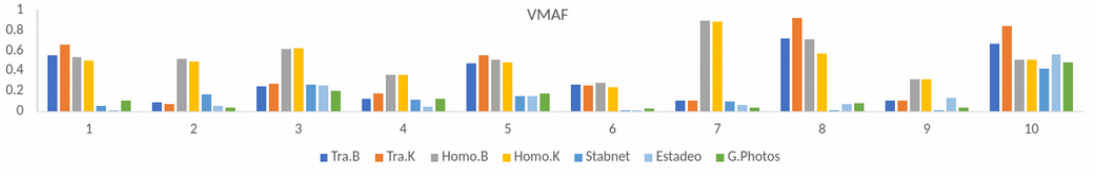


Fig. 7.8 Experimental Results for filter-based DVS (VMAF): video specific results.

using the metrics in our evaluation framework (namely Resolution Preservation, MSE, SSIM, and feature displacement) and VMAF.

Figure 7.9 shows the average experimental results using our evaluation framework for our three systems: Ours-1 and Ours-2 show the results of our pure homography-based systems, while Ours1 and Ours2 show the results of our pure translation-based system. On the other hand, Tra.B and Tra.K show the results of our filter-based approach applied to translation only, while Homo.B and Homo.K show the results of our filter-based approach applied to homographies.

Overall, our pure translation-based systems present better Resolution Preservation results, while the MSE results are similar to all systems, apart from the homography-based results, which present degraded performance when compared to others. The SSIM results, on the other hand, show that the filter-translation-based DVS has better results. For dx and dy, all filter-based results presented better performance, both in the x and y axis, the filter-homography-based being the best between them.

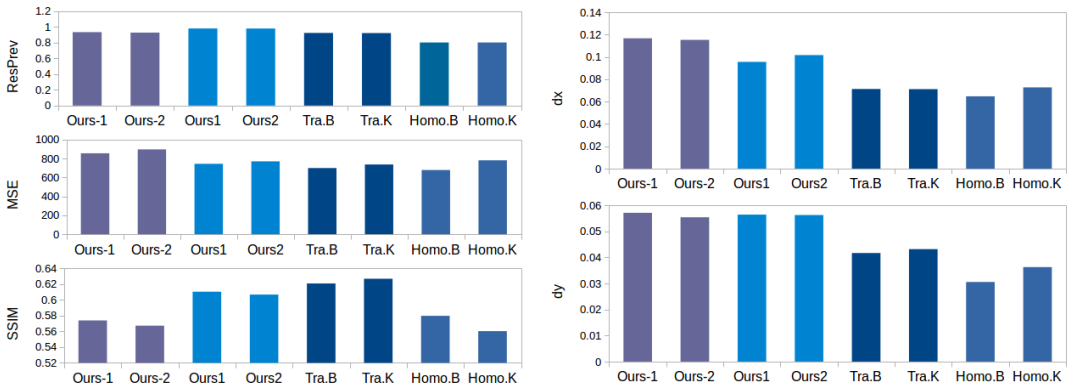


Fig. 7.9 Performance evaluation to assess our proposals: our evaluation framework. The purple bars represent the results obtained with our pure homography-based DVS, while the light blue bars represent the results for our pure translation-based DVS, the navy blue bars represent the results for our filter-translation-based DVS and the mild blue bars represent the results of our filter-homography-based DVS.

Figure 7.10 shows the VMAF results. Our filter-translation-based system provides the best results for this metric. In all metrics, our homography-based does not present comparable results to our other systems. It is expected, since the homography part of the system allows a certain degree of blur and distortion, and using moving averages has not shown comparable results. Hence, we are ought to choose which system, between our translation-based and our filter-based, is the best. While the pure translation-based approach presents the best performance in the Resolution Preservation metric, the filter-based approaches present the best performance in the SSIM (filter-translation), dx/dy (filter-homography), and VMAF (filter-translation) metrics.

Resolution preservation compares how much a frame has been cropped and how much detail has been lost in each frame. On the other hand, the feature displacement is a good measurement of how stable the video is. However, VMAF is an important factor to be taken into account, since it considers the Human Visual System in its calculation. Plus, SSIM results show how similar the output frames are to the ground truth.

Hence, considering the combination of the SSIM, and VMAF, we believe that, even though pure translation-based and filter-homography-based systems present good

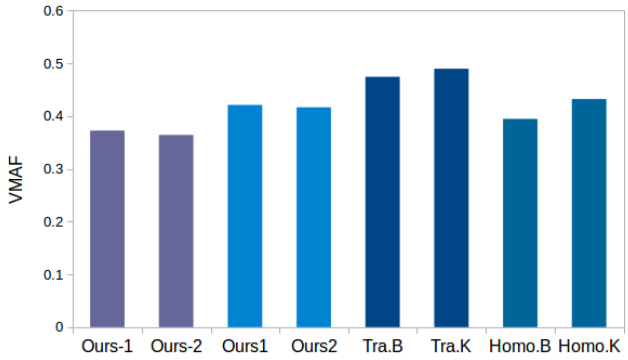


Fig. 7.10 Performance evaluation to evaluate our proposals: VMAF. The purple bars represent the results obtained with our pure homography-based DVS, while the light blue bars represent the results for our pure translation-based DVS, the navy blue bars represent the results for our filter-translation-based DVS and the mild blue bars represent the results of our filter-homography-based DVS.

performance, the filter-tranlation-based approach is the best among the ones we have proposed.

7.5 Proposed Systems and StabNet: a Comparison

In this Section, we consider the possibility of adding our filter-based approaches to our DL-based baseline, StabNet. Although our filter-based approaches and StabNet are significantly different, it is a good approach to evaluate whether they can be combined to provide better stabilization results.

With this in mind, we carry out a performance evaluation using our evaluation framework and VMAF. To obtain the filter-based+StabNet results, we run our filter-based approach on the videos that have been stabilized by StabNet. We run these experiments on a set of 60 videos from our dataset, which contains both unstable and ground truth videos. The experimental results are shown in Figures 7.11 and 7.12, respectively. From left to right, the three different colours show 1) the combination of our filter-based systems and StabNet, 2) our pure filter-based systems, and 3) pure StabNet.

One can observe that our pure filter-based approaches present higher Resolution Preservation than the other approaches. Such result is expected, as the resolution preservation from the videos stabilized by StabNet are initially low, hence the further stabilization by our systems would not be able to provide better results, since information (such as frame sharpness and resolution) has already been lost.

The MSE results show that our pure filter-based systems provide the best results. Since the StabNet's MSE results are high, it shows that our filter-based approach helps improve StabNet's results. However, since the outputs to StabNet tends to be distorted, both because of the homographies performed and the patch stitching the system performs, we believe that our STN does not reliably estimate motion parameters. Also, the output to StabNet tends to be blurry, which tends to deliver higher MSE. Since our filter-based+StabNet results present better MSE results than StabNet, we argue that it is a result of a better stabilization provided by our filter-based approaches.

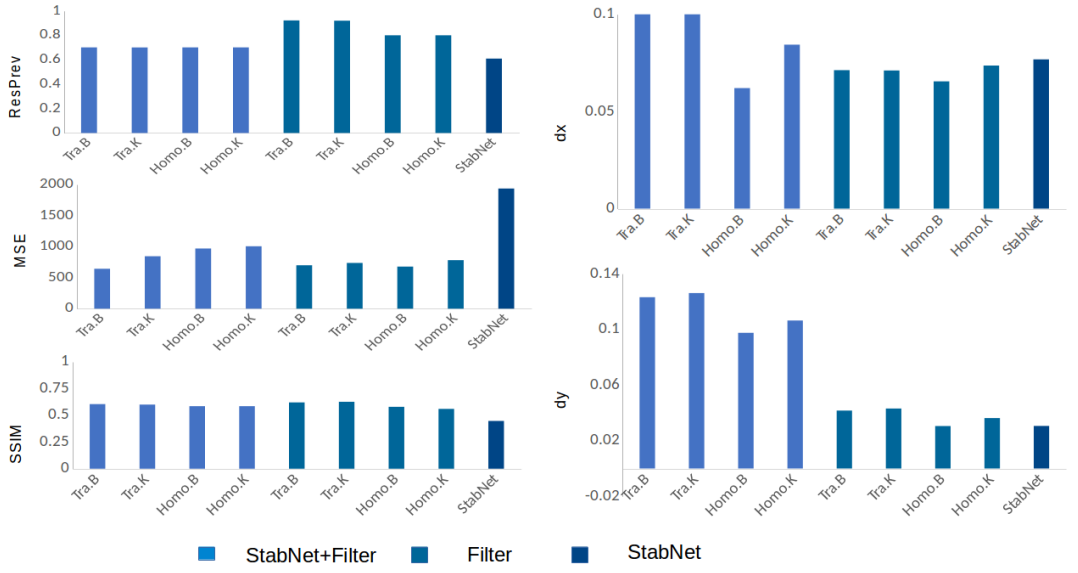


Fig. 7.11 Proposed systems and StabNet: a comparison using our evaluation framework. The light blue bars represent our systems added to StabNet, while the mild blue bars represent our pure systems and the navy blue bars represent pure StabNet.

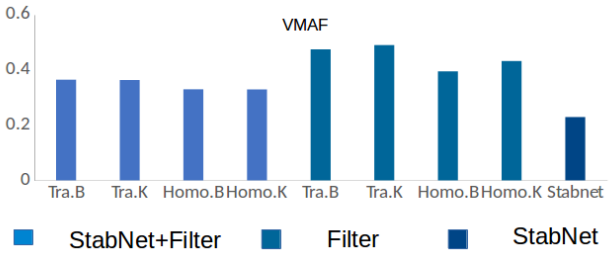


Fig. 7.12 Proposed systems and StabNet: a comparison using VMAF. The light blue bars represent our systems added to StabNet, while the mild blue bars represent our pure systems and the navy blue bars represent pure StabNet.

The SSIM results are similar between our pure filter-based system and our filter-based+StabNet results, with pure StabNet showing the lowest results. It shows that our system helps improve the SSIM from the StabNet output, and that the structure of the frames tend to be preserved. In this case, our systems have managed to improve StabNet's results.

The dx/dy metrics confirm our thoughts from the MSE results. The combined filter-based+StabNet results are higher than both original systems: we believe this result is a cause of low performance of our STN, lead by the initially distorted frames that have been fed to the network. Hence, the combination of both systems present high dx and dy results.

The VMAF results show that, although the filter-based+StabNet outputs have high dx/dy systems, our filter-based approach has helped improve StabNet's VMAF results, which leads us to believe that, overall, the combination is beneficial to StabNet. However, the combination has rather degraded the performance of our systems. Hence, after observing the experimental results, we can conclude that our pure filter-based approaches are still the best option for DVS systems.

7.6 Outcomes

In this Chapter, we have presented a filter-based approach to DVS. Taking into account our homography-based and translation-based approaches, we carry out a performance evaluation comparing a number of filters and parameters for stabilizing videos. We define a stability metric to select the best filters and parameters for each system. Our filter selection results show that the best filters are the Butterworth and Kalman, the former being newly introduced to DVS, to the best of our knowledge. The Butterworth filter has outperformed widely utilized filters, such as Kalman and Gaussian, which shows that there is room for improving camera path smoothing, by possibly experimenting with other alternative filters. We have successfully shown that our filter-based approaches, applied to both the homography and translation cases, outperform our baseline.

The performance evaluation comparing our three proposed systems shows that the filter-translation-based system is the best option between the considered cases. Finally, comparing our filter-based approaches and StabNet to a combination of our filter-based approach and StabNet, we conclude that although the combination improves StabNet results, it tend to present poor performance when compared to our pure filter-based approach.

Chapter 8

Conclusions and Future Developments

The research described in this thesis was a scientific endeavor in improving the currently available Digital Video Stabilization systems. For doing so, we have proposed a novel DVS system that uses STNs, a DL approach to estimate the camera path and moving averages and filters, which are traditional approaches, for smoothing the camera path.

8.1 Summary of Scientific Achievements

The first milestone was a comprehensive video dataset for learning-based video stabilization, which consists of stable-unstable video pairs, with the motion parameters between them, presented in Chapter 3, Section 3.2. This was followed by the development of a Full-Reference evaluation framework, which considers the presence of the ground truth videos, simplifying the evaluation process and avoiding any possible bias, presented in Chapter 3, Section 3.4.

Secondly, we trained and tested an STN for motion estimation between images in Chapter 4. In this step, we utilized the COCO dataset, a popular image dataset. With

this approach, we have simplified the learning process for DVS, by avoiding the need of stable/unstable video pairs.

In a third step, we proposed a number of approaches for video stabilization. Having in mind that the video stabilization problem faces hurdles both in the traditional and in the DL fields, we decided to take advantage of their strengths and simplify the problem. To this end, we leveraged the capabilities of our STN to estimate motion between images and the simplicity of using moving averages or filters to smooth the camera path.

i) Our first approach, presented in Chapter 5, was to use moving averages and full homographies to stabilize videos. We performed a number of motion estimation parameters between the current frame and the previous frames to provide two outputs. The first one consisted of long-term smoothing, and the second, of long and short-term smoothing. Although the output videos outperformed the state of the art, they had added blur due to the interpolation performed during the frame transformation and distortion due to the homography.

ii) To avoid blur and possible distortions caused by the homographies, we proposed a video stabilization system that uses EWMA and moving averages that do not perform full frame transformation, presented in Chapter 6, Section 6.2. Instead, we proposed an approach that performs translations in the x and y axis, so we can avoid interpolating the frames, hence avoiding blur. Because we didn't perform any rotations, we also avoided introducing distortions to the frames. However, this approach allowed some jitter left in the stabilized video, in a tradeoff between video stabilization and overall frame quality. Although our approach presented good performance, we believed that a single camera smoothing strategy was not enough to stabilize videos with the most diverse camera paths.

iii) Having this in mind, we next proposed a filter-based video stabilization in Chapter 7. In this system, we implemented a number of filters (including Kalman and low-pass) and evaluated their performances with a large number of videos, with different camera paths characteristics. With this, we could choose the filters that would provide the best overall performance to the tested videos. We applied our filter-based approach to both our homography-based and translation-based systems.

We carried out our performance evaluation experiments using our evaluation framework, proposed in Chapter 3, and a perceptual metric that has been trained on subjective studies, VMAF, discussed in Chapter 2. Hence, we consider that we have carried out a thorough performance evaluation, since we took into account both objective and subjective metrics.

Then, after proposing three different DVS systems, we compared their performances in Chapter 7, Section 7.4. After careful evaluation, we came to the conclusion that, although the EWMA and filter-based solutions presented similar results, our filter-based DVS system is the best among them, especially the filter-translation-based approach.

Laastly, we have considered the possibility of adding our filter-based approaches to StabNet. We have shown that, although our filters have improved StabNet's results, such improvement was not enough against our pure filter-based approaches, which we considered the best option for DVS.

With our new system, we intend to handle textureless objects, parallax, occlusion, high motion, and to filter unwanted motion while keeping wanted motion. We also intend to avoid blur, distortion, zoom, ghost effect, while being able to accurately smooth the camera path without adding extra unwanted motion.

Indeed, in Chapters 5, 6, and 7 we have been able to prove the robustness of our systems with their effectiveness in a wide range of scenarios, which include textureless objects, parallax, occlusion, and high motion. Since our systems present either damping techniques or low-pass filters, we argue that we have managed to accurately smooth the camera path, by filtering out unwanted motion without adding extra unwanted motion. Although our system in Chapter 5 performs homographies, our system in Chapter 6 perform translation only, which allows us to avoid blur, distortion, and zoom. Also, since our approaches perform either 2D transformations or translation only, we do not add ghost effect, which is a result of 3D DVS systems. In a nutshell, we argue that we have successfully achieved our goals.

8.2 Potential Future Developments

We have presented a number of DVS systems, which present good performance, especially the filter-based approach. However, there is room for complementing our proposal. Our first suggestion is an extension of our filter selection studies. Our studies compared the traditional filters and parameters against three alternative filters. However, there is a wide variety of other potential filters that could be taken into account. Another possibility is a combination of filters.

Another front is the filter selection based on certain camera path profiles. A good question to be asked is if the selected filter and parameter change according to certain camera path characteristics, such as peak frequency or ratio between highest and lowest frequencies. Such a study might lead to a rich discussion based on signal processing and Deep Learning.

However, we are aware that the DL community is working hard towards addressing several different challenges, and we believe that in the future a pure DL-based DVS

approach will be feasible, possibly with the creation of new networks or with the development of novel loss functions, which could address video stability.

References

[1] I. Teivas, "Video event classification using 3d convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2016.

[2] I. W. Selesnick and C. S. Burrus, "Generalized digital butterworth filter design," *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1688–1694, 1998.

[3] "Section 5-4 - standard responses," in *Op Amp Applications Handbook*, W. Jung, Ed. Burlington: Newnes, 2005, pp. 325 – 348.

[4] C. Trudinger, I. Enting, and P. Rayner, "Kalman filter analysis of ice core data 1. method development and testing the statistics," *Journal of Geophysical Research*, vol. 107, 09 2002.

[5] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2014.

[6] M. S. Ito and E. Izquierdo, "A dataset and evaluation framework for deep learning based video stabilization systems," in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019, pp. 1–4.

[7] M. Okade and P. K. Biswas, "Video stabilization using maximally stable extremal region features," *Multimedia Tools and Applications*, vol. 68, no. 3, pp. 947–968, Feb 2014.

[8] D. Shukla, A. Ojha, and R. K. Jha, "A new composite multi-constrained differential-radon warping approach for digital video affine motion stabilization," *Comput. Vis. Image Underst.*, vol. 155, no. C, pp. 83–105, Feb. 2017.

[9] W. Guilluy, L. Oudre, and A. Beghdadi, "Video stabilization: Overview, challenges and perspectives," *Signal Processing: Image Communication*, vol. 90, p. 116015, 2021.

[10] J. Li, T. Xu, and K. Zhang, "Real-time feature-based video stabilization on fpga," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 907–919, April 2017.

[11] S. Liu, M. Li, S. Zhu, and B. Zeng, "Codingflow: Enable video coding for video stabilization," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3291–3302, July 2017.

[12] L. Zhang, Q. Zheng, and H. Huang, "Intrinsic motion stability assessment for video stabilization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 4, pp. 1681–1692, April 2019.

[13] J. Dong and H. Liu, "Video stabilization for strict real-time applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 4, pp. 716–724, April 2017.

[14] S. Cho, J. Wang, and S. Lee, "Video deblurring for hand-held cameras using patch-based synthesis," *ACM Trans. Graph.*, vol. 31, no. 4, Jul. 2012.

[15] S. Liu, B. Xu, C. Deng, S. Zhu, B. Zeng, and M. Gabbouj, "A hybrid approach for near-range video stabilization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1922–1933, Sept 2017.

[16] L. Zhang, Q. Xu, and H. Huang, "A global approach to fast video stabilization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 2, pp. 225–235, Feb 2017.

[17] L. Zhang, X. Q. Chen, X. Y. Kong, and H. Huang, "Geodesic video stabilization in transformation space," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2219–2229, May 2017.

[18] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, Feb 2016.

[19] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 237–246.

[20] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, "Real-time neural style transfer for videos," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 7044–7052.

[21] V. Andrearczyk and P. F. Whelan, "Convolutional neural network on three orthogonal planes for dynamic texture classification," *Pattern Recognition*, vol. 76, pp. 36 – 49, 2018.

[22] Y. Li, G. Wang, L. Nie, Q. Wang, and W. Tan, "Distance metric optimization driven convolutional neural network for age invariant face recognition," *Pattern Recognition*, vol. 75, pp. 51 – 62, 2018, distance Metric Learning for Pattern Recognition.

[23] M. Wang, G. Yang, J. Lin, S. Zhang, A. Shamir, S. Lu, and S. Hu, "Deep online video stabilization with multi-grid warping transformation learning," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2283–2292, May 2019.

[24] S.-Z. Xu, J. Hu, M. Wang, T.-J. Mu, and S.-M. Hu, "Deep video stabilization using adversarial networks," *Computer Graphics Forum*, vol. 37, no. 7, pp. 267–276, 2018.

[25] J. Yu and R. Ramamoorthi, "Robust video stabilization by optimization in cnn weight space," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3795–3803.

[26] J. Choi and I. S. Kweon, "Deep iterative frame interpolation for full-frame video stabilization," *ACM Trans. Graph.*, vol. 39, no. 1, Jan. 2020.

[27] P. Rawat and M. D. Sawale, "Gaussian kernel filtering for video stabilization," in *2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE)*, 2017, pp. 142–147.

[28] J. Sánchez, "Comparison of Motion Smoothing Strategies for Video Stabilization using Parametric Models," *Image Processing On Line*, vol. 7, pp. 309–346, 2017.

[29] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu, "Spatial transformer networks," in *Advances in Neural Information Processing Systems 28*. Curran Associates, Inc., 2015, pp. 2017–2025.

[30] S. Liu, P. Tan, L. Yuan, J. Sun, and B. Zeng, "Meshflow: Minimum latency online video stabilization," in *Computer Vision – ECCV 2016*. Cham: Springer International Publishing, 2016, pp. 800–815.

[31] Y. G. Lee, "Novel video stabilization for real-time optical character recognition applications," *Journal of Visual Communication and Image Representation*, vol. 44, pp. 148 – 155, 2017.

[32] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," *ACM Trans. Graph.*, vol. 31, no. 5, Sep. 2012.

[33] S. Erturk, "Image sequence stabilisation: motion vector integration (mvi) versus frame position smoothing (fps)," in *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, 2001, pp. 266–271.

[34] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11 – 26, 2017.

[35] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, Feb 2015.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[37] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27 – 48, 2016, recent Developments on Deep Big Vision.

[38] Y. H. G. LeCun, Yann; Bengio, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[39] K. Ota, M. S. Dao, V. Mezaris, and F. G. B. D. Natale, "Deep learning for mobile multimedia: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 13, no. 3s, pp. 34:1–34:22, Jun. 2017.

[40] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan 2013.

[41] C.-H. Lin, E. Yumer, O. Wang, E. Shechtman, and S. Lucey, "St-gan: Spatial transformer generative adversarial networks for image compositing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[42] C. Bhagavatula, C. Zhu, K. Luu, and M. Savvides, "Faster than real-time facial alignment: A 3d spatial transformer network approach in unconstrained poses," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 4000–4009.

[43] C.-H. Lin and S. Lucey, "Inverse compositional spatial transformer networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[44] I. Yoo, D. G. C. Hildebrand, W. F. Tobin, W.-C. A. Lee, and W.-K. Jeong, "ssemnet: Serial-section electron microscopy image registration using a spatial transformer network with learned features," in *DLMIA/ML-CDS@MICCAI*, 2017.

[45] B. D. de Vos, F. F. Berendsen, M. A. Viergever, M. Staring, and I. Išgum, "End-to-end unsupervised deformable image registration with a convolutional neural network," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Cham: Springer International Publishing, 2017, pp. 204–212.

[46] M. S. Ito and E. Izquierdo, "Distortion-free video stabilization," in *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2020, pp. 1–4.

[47] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, "Accelerating magnetic resonance imaging via deep learning," in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*, April 2016, pp. 514–517.

[48] K. Ratakonda, "Real-time digital video stabilization for multi-media applications," in *Circuits and Systems, 1998. ISCAS '98. Proceedings of the 1998 IEEE International Symposium on*, vol. 4, May 1998, pp. 69–72 vol.4.

[49] M. Grundmann, V. Kwatra, and I. Essa, "Auto-directed video stabilization with robust l1 optimal camera paths," in *CVPR 2011*, June 2011, pp. 225–232.

[50] S. Liu, L. Yuan, P. Tan, and J. Sun, "Bundled camera paths for video stabilization," *ACM Trans. Graph.*, vol. 32, no. 4, Jul. 2013.

[51] 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*, June 2014, pp. 4209–4216.

[52] Yasuyuki Matsushita, Eyal Ofek, Xiaoou Tang, and Heung-Yeung Shum, "Full-frame video stabilization," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, June 2005, pp. 50–57 vol. 1.

[53] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung, "Video stabilization using robust feature trajectories," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 1397–1404.

[54] D. Shukla, R. K. Jha, and A. Ojha, "Digital image stabilization using similarity transformation over constrained differential-radon warping vectors," *Signal Processing: Image Communication*, vol. 47, pp. 115 – 130, 2016.

[55] B.-Y. Chen, K.-Y. Lee, W.-T. Huang, and J.-S. Lin, "Capturing intention-based full-frame video stabilization," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1805–1814, 2008.

[56] A. Litvin, J. Konrad, and W. C. Karl, "Probabilistic video stabilization using Kalman filtering and mosaicing," in *Image and Video Communications and Processing 2003*, B. Vasudev, T. R. Hsing, A. G. Tescher, and T. Ebrahimi, Eds., vol. 5022, International Society for Optics and Photonics. SPIE, 2003, pp. 663 – 674.

[57] G. Puglisi and S. Battiato, "A robust image alignment algorithm for video stabilization purposes," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1390–1400, Oct 2011.

[58] Y. Matsushita, E. Ofek, Weina Ge, Xiaoou Tang, and Heung-Yeung Shum, "Full-frame video stabilization with motion inpainting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 7, pp. 1150–1163, July 2006.

[59] H.-C. Chang, S.-H. Lai, and K.-R. Lu, "A robust real-time video stabilization algorithm," *Journal of Visual Communication and Image Representation*, vol. 17, no. 3, pp. 659 – 673, 2006, special Issue on Real-Time Imaging.

[60] F. Zhang, J. Wang, H. Zhao, R. R. Martin, and S. Hu, "Simultaneous camera path optimization and distraction removal for improving amateur video," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5982–5994, Dec 2015.

[61] C. Liang, L. Chang, and H. H. Chen, "Analysis and compensation of rolling shutter effect," *IEEE Transactions on Image Processing*, vol. 17, no. 8, pp. 1323–1330, Aug 2008.

[62] Z.-Q. Wang, L. Zhang, and H. Huang, "Multiplane video stabilization," *Computer Graphics Forum*, vol. 32, no. 7, pp. 265–273, 2013.

[63] S. Battiato, G. Gallo, G. Puglisi, and S. Scellato, "Sift features tracking for video stabilization," in *14th International Conference on Image Analysis and Processing (ICIAP 2007)*, 2007, pp. 825–830.

[64] S. Kim, S. Kang, T. Wang, and S. Ko, "Feature point classification based global motion estimation for video stabilization," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 1, pp. 267–272, 2013.

[65] T. Su, Y. Nie, Z. Zhang, H. Sun, and G. Li, "Video stitching for handheld inputs via combined video stabilization," in *SIGGRAPH ASIA 2016 Technical Briefs*, ser. SA '16. New York, NY, USA: Association for Computing Machinery, 2016.

[66] F. Hu, J. Ma, L. Shen, and H. Du, "Digital video stabilization based on multilayer gray projection," *Signal Processing: Image Communication*, vol. 68, pp. 42 – 57, 2018.

[67] B.-H. Chen, A. Kopylov, S.-C. Huang, O. Seredin, R. Karpov, S.-Y. Kuo, K. R. Lai], T.-H. Tan, M. Gochoo, D. Bayanduuren, C.-S. Gong, and P. C. Hung, "Improved global motion estimation via motion vector clustering for video stabilization," *Engineering Applications of Artificial Intelligence*, vol. 54, pp. 39 – 48, 2016.

[68] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510–517.

[69] J. Dong, Y. Xia, Q. Yu, A. Su, and W. Hou, "Instantaneous video stabilization for unmanned aerial vehicles," *Journal of Electronic Imaging*, vol. 23, no. 1, pp. 1 – 10, 2014.

[70] L. Kejriwal and I. Singh, "A hybrid filtering approach of digital video stabilization for uav using kalman and low pass filter," *Procedia Computer Science*, vol. 93, pp. 359 – 366, 2016, proceedings of the 6th International Conference on Advances in Computing and Communications.

[71] W. A. . A. A. Walha, A., "Video stabilization with moving object detecting and tracking for aerial video surveillance," *Multimed Tools Appl*, 2015.

[72] L. Araneda and M. Figueroa, "Real-time digital video stabilization on an fpga," in *2014 17th Euromicro Conference on Digital System Design*, Aug 2014, pp. 90–97.

[73] T. Yabuki and Y. Yamaguchi, "Real-time video stabilization on an fpga," in *INTERNATIONAL CONFERENCE ON SMART STRUCTURES AND SYSTEMS - ICSSS'13*, March 2013, pp. 114–119.

[74] G. Li, "FPGA implementation of real-time digital image stabilization," in *Selected Papers from Conferences of the Photoelectronic Technology Committee of the Chinese Society of Astronautics: Optical Imaging, Remote Sensing, and Laser-Matter Interaction 2013*, vol. 9142, International Society for Optics and Photonics. SPIE, 2014, pp. 509 – 515.

[75] F. Liu, M. Gleicher, H. Jin, and A. Agarwala, "Content-preserving warps for 3d video stabilization," in *ACM SIGGRAPH 2009 Papers*, ser. SIGGRAPH '09. New York, NY, USA: ACM, 2009, pp. 44:1–44:9.

[76] Z. Zhou, H. Jin, and Y. Ma, "Plane-based content preserving warps for video stabilization," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[77] C. Buehler, M. Bosse, and L. McMillan, "Non-metric image-based rendering for video stabilization," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, Dec 2001, pp. II–II.

[78] S. Liu, Y. Wang, L. Yuan, J. Bu, P. Tan, and J. Sun, "Video stabilization with a depth camera," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 89–95.

[79] T. H. Lee, Y. gu Lee, and B. C. Song, "Fast 3d video stabilization using roi-based warping," *Journal of Visual Communication and Image Representation*, vol. 25, no. 5, pp. 943 – 950, 2014.

[80] F. Liu, M. Gleicher, J. Wang, H. Jin, and A. Agarwala, "Subspace video stabilization," *ACM Trans. Graph.*, vol. 30, no. 1, pp. 4:1–4:10, Feb. 2011.

[81] M. Grundmann, V. Kwatra, D. Castro, and I. Essa, "Calibration-free rolling shutter removal," in *2012 IEEE International Conference on Computational Photography (ICCP)*, April 2012, pp. 1–8.

[82] Y. Wang, F. Liu, P. Hsu, and T. Lee, "Spatially and temporally optimized video stabilization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1354–1361, Aug 2013.

[83] F. Liu, Y. Niu, and H. Jin, "Joint subspace stabilization for stereoscopic video," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[84] M. Irani, "Multi-frame correspondence estimation using subspace constraints," *International Journal of Computer Vision*, 2002.

[85] J. Kopf, "360° video stabilization," *ACM Trans. Graph.*, vol. 35, no. 6, Nov. 2016.

[86] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255.

[87] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[88] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[89] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[90] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[91] Y. Nakajima and H. Saito, "Robust camera pose estimation by viewpoint classification using deep learning," *Computational Visual Media*, vol. 3, no. 2, pp. 189–198, Jun 2017.

[92] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 568–576.

[93] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *Advances in Neural Information Processing Systems 29*. Curran Associates, Inc., 2016, pp. 91–99.

[94] Z. Liu, R. A. Yeh, X. Tang, Y. Liu, and A. Agarwala, "Video frame synthesis using deep voxel flow," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[95] T. Hyun Kim, K. Mu Lee, B. Scholkopf, and M. Hirsch, "Online video deblurring via dynamic temporal blending network," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[96] M. Zhao and Q. Ling, "Pwstablenet: Learning pixel-wise warping maps for video stabilization," *IEEE Transactions on Image Processing*, vol. 29, pp. 3582–3595, 2020.

[97] L. Zhang, Q. Zheng, H. Liu, and H. Huang, "Full-reference stability assessment of digital video stabilization based on riemannian metric," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6051–6063, 2018.

[98] A. Goldstein and R. Fattal, "Video stabilization using epipolar geometry," *ACM Trans. Graph.*, vol. 31, no. 5, Sep. 2012.

[99] Y. J. Koh, C. Lee, and C. Kim, "Video stabilization based on feature trajectory augmentation and selection and robust mesh grid warping," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5260–5273, 2015.

[100] J. Yu, , and R. Ramamoorthi, "Selfie video stabilization," *ECCV*, 2018.

[101] H. Qu, L. Song, and G. Xue, "Shaking video synthesis for video stabilization performance assessment," in *2013 Visual Communications and Image Processing (VCIP)*, 2013, pp. 1–6.

[102] S. Lu, J. You, B. Ceulemans, M. Wang, and A. Munteanu, "Synthesis of shaking video using motion capture data and dynamic 3d scene modeling," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 1438–1442.

[103] W. Guilluy, A. Beghdadi, and L. Oudre, "A performance evaluation framework for video stabilization methods," in *2018 7th European Workshop on Visual Information Processing (EUVIP)*, 2018, pp. 1–6.

[104] C. Zhang, P. Chockalingam, A. Kumar, P. Burt, and A. Lakshmikumar, "Qualitative assessment of video stabilization and mosaicking systems," in *2008 IEEE Workshop on Applications of Computer Vision*, 2008, pp. 1–6.

[105] B. Zhai, J. Zheng, Y. Wang, and C. Zhang, "A multi-scale evaluation method for motion filtering in digital image stabilization," in *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, 2015, pp. 682–688.

[106] W. Guilluy, A. Beghdadi, and L. Oudre, "A performance evaluation framework for video stabilization methods," in *2018 7th European Workshop on Visual Information Processing (EUVIP)*, 2018, pp. 1–6.

[107] M. Souza and H. Pedrini, "Visual rhythms for qualitative evaluation of video stabilization," in *EURASIP Journal on Image and Video Processing*, 2018.

[108] R. S. . B. M. Sun, D., "A quantitative analysis of current practices in optical flow estimation and the principles behind them," *International Journal of Computer Vision*, vol. 106.

[109] C. Morimoto and R. Chellappa, "Evaluation of image stabilization algorithms," in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, vol. 5, 1998, pp. 2789–2792 vol.5.

[110] Z. Cui and T. Jiang, "No-reference video shakiness quality assessment," in *Computer Vision – ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham: Springer International Publishing, 2017, pp. 396–411.

[111] M. Niskanen, O. Silven, and M. Tico, "Video stabilization performance assessment," in *2006 IEEE International Conference on Multimedia and Expo*, 2006, pp. 405–408.

[112] C. Chang, C. Chou, and E. Y. Chang, "Clkn: Cascaded lucas-kanade networks for image alignment," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 3777–3785.

[113] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep image homography estimation," *CoRR*, vol. abs/1606.03798, 2016.

[114] F. E. Nowruzi, R. Laganiere, and N. Japkowicz, "Homography estimation from image pairs with hierarchical convolutional networks," in *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, 2017, pp. 904–911.

[115] J. Fan, X. Cao, Z. Xue, P.-T. Yap, and D. Shen, "Adversarial similarity network for evaluating image alignment in deep learning based registration," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger, Eds. Cham: Springer International Publishing, 2018, pp. 739–746.

[116] G. Huang, M. Mattar, H. Lee, and E. G. Learned-miller, "Learning to align from scratch," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 764–772.

[117] M. Kowalski, J. Naruniec, and T. Trzcinski, "Deep alignment network: A convolutional neural network for robust face alignment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[118] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarfuglia, "Exploring representation learning with cnns for frame-to-frame ego-motion estimation," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 18–25, 2016.

[119] S. Wang, R. Clark, H. Wen, and N. Trigoni, "Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 2043–2050.

[120] N. Yang, R. Wang, J. Stuckler, and D. Cremers, "Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[121] G. Iyer, J. Krishna Murthy, G. Gupta, M. Krishna, and L. Paull, "Geometric consistency for self-supervised end-to-end visual odometry," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[122] B. Cîrstea and L. Likforman-Sulem, "Tied spatial transformer networks for digit recognition," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016, pp. 524–529.

[123] V. Mnih, N. Heess, A. Graves, and k. kavukcuoglu, "Recurrent models of visual attention," in *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., 2014, pp. 2204–2212.

[124] C. C. Christopher J.C. Burges Yann LeCun, "The mnist database of handwritten digits," b/mnist/, accessed: 2020-04-30.

[125] S. Kim, S. Lin, S. R. Jeon, D. Min, and K. Sohn, "Recurrent transformer networks for semantic correspondence," in *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018, pp. 6126–6136.

[126] A. García, J. A. García, and L. M. Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, pp. 158 – 165, 2018.

[127] Y. Fang, S. Gao, J. Li, W. Luo, L. He, and B. Hu, "Multi-level feature fusion based locality-constrained spatial transformer network for video crowd counting," *Neurocomputing*, 2020.

[128] K. S. Erer, "Adaptive usage of the butterworth digital filter," *Journal of Biomechanics*, vol. 40, no. 13, pp. 2934 – 2943, 2007.

[129] G. Welch and G. Bishop, "An introduction to the kalman filter," 1995.

[130] "ffmpeg," https://www.ffmpeg.org/, accessed: 2010-10-30.

[131] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[132] S. B. Balakirsky and R. Chellappa, "Performance characterization of image stabilization algorithms," *Real-Time Imaging*, vol. 2, no. 5, pp. 297 – 313, 1996.

[133] X. Li, Q. Guo, and X. Lu, "Spatiotemporal statistics for video quality assessment," *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3329–3342, 2016.

[134] C. G. Bampis, Z. Li, and A. C. Bovik, "Spatiotemporal feature integration and model fusion for full reference video quality assessment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 8, pp. 2256–2270, 2019.

[135] S. Li, F. Zhang, L. Ma, and K. N. Ngan, "Image quality assessment by separately evaluating detail losses and additive impairments," *IEEE Transactions on Multimedia*, vol. 13, no. 5, pp. 935–949, 2011.

[136] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

[137] "Google photos," https://www.google.com/photos/, accessed: 2019-10-30.

[138] M. R. Souza and H. Pedrini, "Digital video stabilization based on adaptive camera trajectory smoothing," *EURASIP Journal on Image and Video Processing*, 2018.