Video stabilization: overview, challenges and perspectives

Wilko Guilluy[a], Laurent Oudre[a,*], Azeddine Beghdadi[a]

[a] *Université Sorbonne Paris Nord, L2TI, UR 3043, F-93430, Villetaneuse, France.*

Abstract

Video Stabilization (VS) has been an active area of research in the last two decades. Many approaches have been successfully proposed and it is time to take a step back and offer a glimpse and a critical look to this hot topic. Among the questions that should be answered : "is VS a solved problem"?. Is there still room for further improvement and at which level and how? The main purpose of this contribution is to answer such questions and to provide a fairly unifying framework to allow a better understanding of the progress of this research subject with appreciable industrial and academic benefits. In this paper we focus on the main challenges, practical aspects and mathematical core concepts of the video stabilization techniques. We also put a special attention to the Video Stabilization Quality Assessment (VSQA) by introducing new methodology inspired by the research results on Image Quality Assessment (IQA) in its broad sense. Finally we also discuss some new research directions to overcome the limitations of the existing methods.

Keywords: video stabilization, video processing

Contents

*Corresponding author

1. Introduction

The continuous development of video sensors and their miniaturization has extended their use in various applications ranging from video surveillance systems to computer-assisted surgery and the analysis of physical and astronomical phenomena [1]. Nowadays it becomes possible to capture video sequences in any environment and without any heavy and complex adjustments as was the case with the old video acquisition sensors [2]. However, the acquired visual information still suffers from some annoying distortions. One of the most perceptually annoying degradation is related to the image instability due to camera movement during the acquisition [3]. This source of degradation manifests as uncontrolled oscillations of the whole frames and may be accompanied with a blurring effect. This affects the perceptual image quality and produces visual discomfort.

These instabilities can produce different types of degradation, such as abrupt motion, that may occur when using hand-held devices, or high-frequency tremors, such as those due to particular standing or moving postures maintained while filming the scene. This can cause important visual discomfort [4, 5]. Lower-frequency motion, such as the up and down movements resulting from walking while filming, can distract the viewer from the focus of the video [6]. Finally, for a camera equipped with a rolling shutter sensor, fast camera movements can induce geometrical deformations on the captured scene [7, 8]. Digital stabilization aims at creating a new video with the same visual content but without these unintentional motion components.

Professional videos are captured using mechanical stabilizers such as tripods or dollies [9] to enforce carefully planned camera movements in order to reduce such undesirable effects. There also exist some hardware solutions such as electronic image stabilizers or gyroscope based technologies that prevent video from blurriness and oscillations [10]. While these hardware solutions produce satisfying results, they fail in some cases, are device dependent and are not widely available. As a result, most amateur videos contain unintended camera movements [11]. In this context, the use of software tools seems to be the most promising solution. The main reasons are the flexibility, the ease of use and the possibility to update and adapt the software solutions to various environments and applications. Furthermore, they offer the advantage of being applicable to older videos and could be of great importance for cultural heritage video restoration applications [12]. This article focuses on these software solutions, often referred to as Digital Video Stabilization (DVS). The developed solutions aim at removing or at least reducing instabilities that mainly manifest themselves as abnormal involuntary/voluntary camera movements. Digital video stabilization is useful in various contexts. As the production and diffusion of video increases, the facilitation of high-quality amateur videos becomes an important field for video-sharing platforms such as Youtube [9]. In professional contexts, law enforcement agencies have increasingly access to videos taken on the spot as evidence. Similarly, they increasingly make use of body cameras, which suffer from sever shakes whenever the wearer is running [13]. Video surveillance cameras can also suffer from detrimental jitter, often due to meteorological conditions [14]. Stabilizing such videos can make their exploitation much easier.

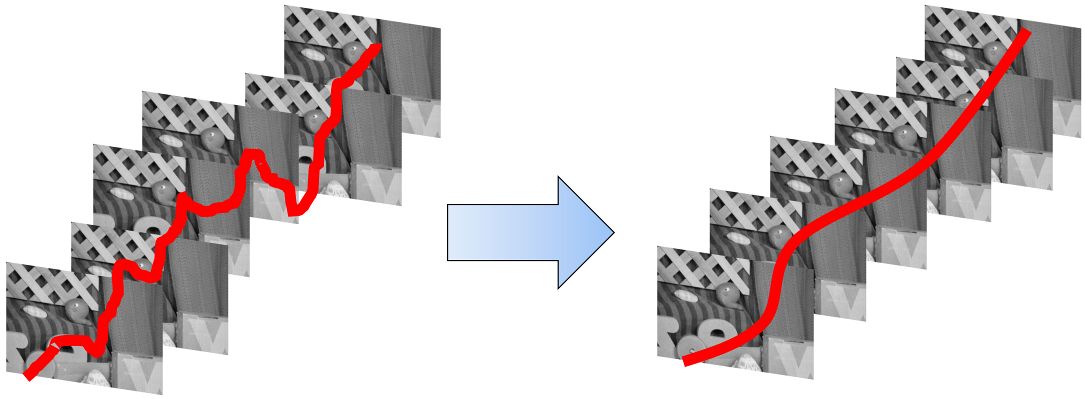

Figure 1: Principle of video stabilization

Other application domains that can benefit from this technology are the medical imaging such as video-guided surgery [15], or remote control of unmanned aerial vehicles [16]. Video stabilization also allows the separation of camera-induced motion and object-dependent motion. This can serve as a pre-processing step in many video analysis processes that use object motion, such as background substraction or visual tracking [14, 17].

While digital stabilization can use additional information from gyroscopes or accelerometers [4], or different viewpoints [18] to improve or facilitate the process, most methods only rely on the video sequence taken from a single camera. Early methods use simple 2-dimensional models such as translations or similarities to represent the camera motion and remove all perceived camera motion to obtain a video corresponding to a simulated video captured by a fixed virtual camera [19]. Motion filters and path-fitting techniques have since been introduced to take intentional camera motion into account, in order to simulate professional camera movements [2]. Similarly, more complex motion models, based on structure-from-motion methods, have been proposed. These computationally demanding solutions, relying on 3-dimensional models, become now attractive and practical thanks to the current high-performance computing (HPC) technologies [20]. However, computing depth from a video sequence remains a long and difficult process that fails in many situations, hence the enduring popularity of 2-dimensional models. Another type of models, that attempts to obtain visually-plausible rather than physically-accurate videos, has emerged more recently [21].

The intend of this paper is not to list all the existing VS methods and topics related to this field of research, but rather to offer a structured and detailed overview by focusing on the most representative approaches developed during the last two decades. Highlighting some limitations on the VS process itself as well as the lack of an accepted methodology for comparing the huge number of developed techniques is another major goal of this contribution. In the following, we conduct such overview by using an incremental approach based on the essential steps of any DVS method. Following this approach, this article outlines and discusses the different components of DVS as well as the methods for evaluating the stabilization results.

The main contributions of this paper are the following

- To provide a comprehensive and critical overview of VS methods;
- To discuss the current challenges in VS and Video Stabilization Quality Evaluation (VSQA);
- To open new perspectives in both VS and VSQA.

This article is organized as follows: Section 2 presents some basic notions of video stabilization and the structure of this overview. Section 3 details the different components and approaches used for video stabilization. Section 4 presents several aspects related to the performances of VS methods, including an overview of video stabilization quality assessment (VSQA) methods. Finally, Section 5 summarizes the challenges and perspectives of digital stabilization. Two summary tables are displayed at the end of the manuscript : Table 1 describes the main chronological approaches for VS along with their advantages and limitations, while Table 2 displays the main available datasets for the evaluation of VS methods.

2. Basic notions on video stabilization

Video stabilization (Figure 1) aims at transforming a video I corrupted by involuntary camera movements into a stabilized video $\tilde{I}$, in which these movements are smoothed in order to produce a coherent and continuous video stream with low visual discomfort and better display quality. This operation is a complex process composed of many steps that could be roughly grouped into two main block as illustrated in Figure 2. First, the original video I is analyzed through motion estimation process. The aim of this first phase is to compute estimates of the camera movements from the video. In a second phase, these estimated camera movements are corrected and smoothed, as in attempt to remove their involuntary parts while preserving their voluntary parts. The video

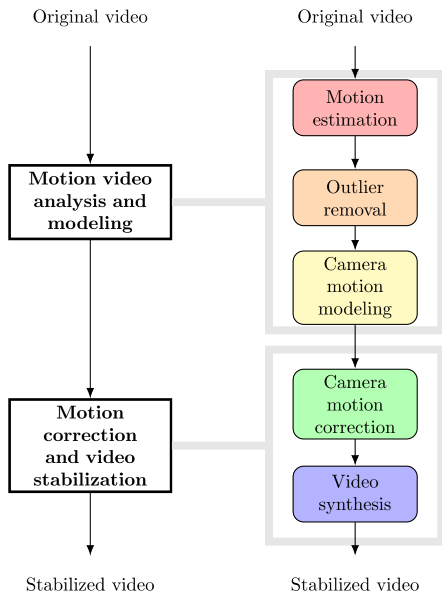

Figure 2: Main steps for video stabilization

is finally processed by using the softened camera movements, so as to generate the stabilized video signal $\tilde{I}$.

As mentioned in Section 1, video stabilization has been a major field of research during the last two decades [22–24] due to the wide range of its potential applications [16, 25–28] . Many approaches have been introduced in the literature to solve this problem. Although based on these two general principles (analysis and correction), the methods have evolved by adding pre/post processing steps and using more precise models. In particular, the level of complexity in the video analysis step has increased across the years in order to refine the estimation of the camera motion. As a result, most of current state-of-the-art methods are composed of five to ten processing blocks that are conceived to adapt to the different situations encountered throughout the stabilization process. In this article, we provide a structured overview of the stabilization methods proposed in the literature, according to the chart-flow presented on Figure 2. To this end, we propose to decompose each of the two main stages of video stabilization into a series of functional blocks, that will be studied and described individually. Although all these blocks are not necessary present in all published methods, they constitute a convenient way to compare the different approaches according to the same analysis grid.

The first stage of video stabilization consists in analysing the whole observed video in order to detect and identify the undesirable motion sources. The aim is then to estimate the camera motion and discriminate it from all the motions in the acquired scene. In the following, we decompose this analysis stage into three main successive steps : motion estimation, outliers removal and camera motion modeling. First, the frames of the video are analyzed so as to understand all the movements in the video : this is the **motion estimation** step. These movements might be due to the camera or to moving objects/subjects in the scene. In order to only focus on the movements that are in fact due to the camera, the second step consists in **outlier removal or motion outlier detection/removal**. Based on general assumptions on the possible impacts of camera movements on the video, this block discards all perceptually inconsistent movements that are be used in the stabilization process. Finally, the last block is dedicated to the **camera motion modeling** from the video. As will be seen in the next section, this can be done either by assuming a geometrical model of the camera displacement (in this case the block outputs geometrical parameters of the camera), or by using an empirical model which does not take the geometrical constraints into account.

At the end of the first stage, camera motion estimates are available and can be used to process the shaky video. The second stage of video stabilization consists in the processing of the video. More specifically, the camera motion models output from the first stage goes through a correction process so as to reconstruct a more visually pleasing video. In the following, we decompose this processing stage into two main successive blocks : camera motion correction and video synthesis. The first block performs a **camera motion correction** by applying a low-pass filter in order to smooth the camera movements. Whether geometrical parameters of the camera are available or not, the correction step aims at suppressing the involuntary camera movements and to compute a new plausible camera motion. In this step, the strength of the stabilization can also be adjusted so as to provide pleasing results for the viewer. Finally, the new camera movements are applied back to the original disturbed video, within the **video synthesis** step. This final step reconstructs a new video using the smoothed camera movements.

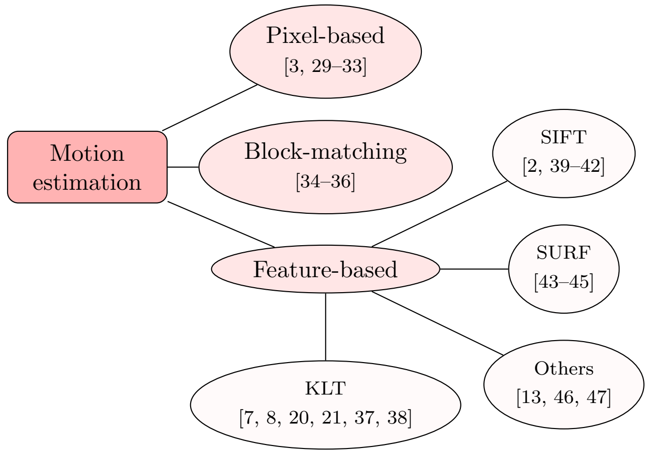

Figure 3: Main approaches for motion estimation

3. Comprehensive overview of video stabilization methods

We provide here step-by-step a comprehensive overview of the video stabilization pipeline by describing each of the different components in a sequential order. First, video analysis estimates movements present in the video (Section 3.1), selects those resulting from the motion of the camera (Section 3.2) and uses them to derive the original trajectory of the camera (Section 3.3). The video is then stabilized by modifying the estimated camera trajectory (Section 3.4) and re-synthesising the video (Section 3.5), by using this corrected trajectory so as to make it consistent with a simulated scene as captured by a virtual camera following this corrected trajectory.

The aim of the whole process is to transform a video sequence I containing unintentional camera movements into a stabilized sequence $\tilde{I}$. In the following, let $\mathbf{z}_t \triangleq [x_t, y_t, 1]$ be a pixel belonging to frame t and $I_t(\mathbf{z}_t)$ the luminance channel of the frame t at the pixel $\mathbf{z}_t$.

3.1. Motion estimation

The first step of any video stabilizer is to estimate the camera movements. Motion estimation aims to recover movements present in the video sequence. The camera motion parameters are then estimated by analyzing the spatio-temporal correspondence between consecutive frames. Pixels (or blocks of pixels) from the first frame are matched with the pixels (or blocks of pixels) in the following frame using a similarity measure or a distance metric. Several approaches have been introduced [48–50] to solve this problem: some try to find a match for every pixel in the frame (Section 3.1.1), others use blocks of pixels (Section 3.1.2) and finally, points of interest can be used to estimate the movements (Section 3.1.3).

3.1.1. Pixel-based matching

Pixel-based matching methods aim at determining the motion of pixels between two frames. Each pixel of the first frame corresponds to the projection of a 3D point in the observed scene onto the camera plane, and is matched with the pixel of the following frame corresponding to the projection of the same 3D point onto the new camera plane. To determine these correspondences, the luminance of any given object is assumed to be constant throughout two consecutive frames of the observed video sequence. However, this corresponds to an ill-posed problem as many pixels in a given pair of frames could have similar luminance. Therefore, to solve this problem additional constraints are needed to obtain a unique solution.

Early methods suppose that the movements are predominantly caused by the motion of the camera , which is modeled by a 2D transformation. Such methods, instead of computing the displacement for every pixel, search for the transformation model that gives the best fit of the overall displacement within the frame. The model parameters are easily estimated by minimizing the luminance difference between the two adjacent frames [29]. If H_t denotes the transformation matrix between the frames t and $t+1$, the optimum parameters minimize the differences $||I_t(H_t\mathbf{z}_t) - I_{t+1}(\mathbf{z}_{t+1})||^2$ for all pixels. Robust functions have also been used to make this approach more robust to outliers [30]. This approach saves time, but needs a pre-determined model and cannot be used with an outlier removal scheme. It is also sensitive to illumination changes.

Another approach is to determine the optical flow between adjacent frames. Optical flow consists in finding the displacement field (u_t, v_t) of each pixel $\mathbf{z}_t$ between two consecutive frames I_t and I_{t+1} . These displacements are estimated thanks to several assumptions, such as local motion similarity [51] or piece-wise smoothness to the flow field [52]. The flow is often computed using the spatial and temporal gradients following an iterative scheme [53]. The main advantage of using optical flow is that it recovers a dense flow field, which is necessary for some stabilization methods [31] [32]. However, computing dense optical flow is computationally demanding. This can be partially alleviated by using other faster matching methods to compute an initial flow before running the iterative algorithm [33, 54]. Another option, which has been used in real-time applications [3], is to use sparse optical flow, which do not solve for motion in low-gradient areas. Liu et al. [31]

report 1.1 seconds per frame to compute the optical flow out of 1.5 seconds per frame for the whole stabilization algorithm. However, despite all these efforts, the main drawback of optical flow remains the computational load.

3.1.2. Block-matching

Instead of finding correspondences between pixels, block-matching methods use blocks of pixels and estimate their displacements between consecutive frames. The use of blocks allows to remove some ambiguities that may occur when matching individual pixels, by decreasing the chance of several matches being detected. Furthermore, by assuming that motion between two frames is limited, it is possible to consider only blocks of pixels within a certain distance from the block to be matched in order to avoid over-matching. To that end, block-matching approaches are based on search windows that constrain the plausible motions. Block-matching approach offers a flexible trade-off between complexity, computational efficiency and accuracy. However, it is limited by some drawbacks such as the aperture and correspondence problem . A lot of works has been done to overcome this kind of drawbacks [55, 56].

3.1.3. Feature-matching

Feature matching seeks to identify points in the scene that are easily recognizable. In this case, only the displacements of these points of interest are computed. By processing the entire video frame by frame, the positions of these points of interest can be tracked using the properties of the selected features, forming trajectories. One of the advantages of using this approach is that the same point can be tracked and recognized across many frames. In particular, the study of the trajectories can give a better insight on the movements present in the scene. Many features detection algorithms have been proposed in the literature. The Kanade–Lucas–Tomasi (KLT) feature tracker [57] has been used to track features across videos in several methods [8, 16, 20, 21, 37, 58]. A feature detection algorithm such as is used to initialize the position of tracked points. Harris corners operator is one of this kind of features detector used for this purpose [14]. These features are then tracked using optical flow technique. This is followed by a checking process to assess the accuracy of the tracking phase. The features' trajectory is then ended when a track failure occurred. To this end, scale pyramids can be used to detect high-amplitude movements more easily [7].

SIFT points are also widely used [2, 39–42]. These features use descriptors based on the image gradient to obtain very specific descriptors that make matches very reliable. These descriptors

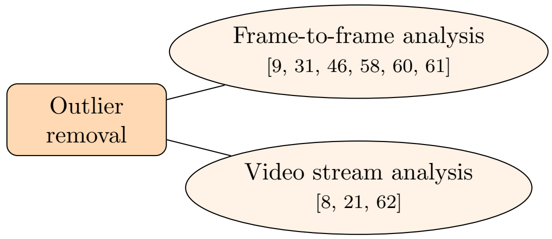

Figure 4: Main approaches for outlier removal

contain the orientation of the feature in order to be rotation-invariant, and detection is used at several image scales that help avoid problems caused by zooming, although it is slower than most alternatives. SURF points were designed on similar principles [59], but optimized for speed, making them a good alternative [43–45]. SIFT features have also been used in conjunction with line detection methods [38], as deformations caused by stabilization are particularly visible on lines.

Other interesting features like Maximally Stable Extremal Regions (MSER) [46] or FAST corners using BRIEF descriptors [13, 47] have been succesfully used.

Feature-matching provides accurate and fast results, and the obtained trajectories allow for additional temporal analysis in the remaining steps of the process, although scenes with large uniform regions can sometimes yield few features per frame. This is one of the limitations of this kind of feature-matching methods.

3.2. Outlier removal

It is worth noticing that while all movements observed in the video sequence are affected by the motion of the camera, only a fraction could be suitable to determine the effective camera motion. Indeed, the presence of moving objects in the filmed scene can be a source of errors, making the discrimination between the various motions (camera / objects) rather a difficult task. Errors can also occur while determining the movements in the video sequence. Finally, some movements may be too complex for a given motion model. Detecting and removing such movements is important to ensure accurate camera motion analysis. Therefore, several methods use a post-processing step to remove these outliers. Two main approaches can be used, that are either based on frame-to-frame analysis (Section 3.2.1) or on the whole video stream (Section 3.2.2).

3.2.1. Frame-to-frame analysis

Most outlier detectors consider two adjacent frames and label as outliers all the displacements that do not fit the general observed movement. This can be done by computing the fitting error between the estimated camera model and the individual movements in the video. If the majority of movements are caused solely by the camera motion, they are assumed to fit the camera motion model, and those deviating from the model are considered unreliable.

The most commonly used method is the RANSAC algorithm [63]. It is based on a motion model and from observed data it seeks to estimate the true motion model parameters. RANSAC randomly selects data to determine the model parameters and measures the distance between the expected positions and the observed positions, with a threshold determining whether a given point is considered inlier or outlier. The parameters resulting in the fewest outliers are then selected. Different variants on RANSAC have been used such as umLESAC [60] or ORSA [61]. These approaches have the advantage of estimating camera motion and detecting outliers at the same time. But, the estimation could be biased when the majority assumption is not satisfied for a single frame.

Assumptions on camera motion can also be used to determine outliers. Spurious movements can be removed by thresholding the flow field according to velocity [46], smoothness [31] or spatial constraints [9, 58].

3.2.2. Video stream analysis

Outlier detection can also take into account motion over more than two frames. This allows to analyze the evolution of motion vectors over time. However it requires tracking points over several frames. Trajectories recovered using feature point tracking is often used in this regard [64].

One criterion that can be exploited is the difference between expected and observed motion. In particular, the motion induced by the camera can be modeled as a projection onto a low rank subspace. Therefore, trajectories whose projections differ strongly from the original motion at any given time are considered erroneous and discarded [21]. A second criterion is related to the duration of the trajectory. Since moving objects often leave the frame quickly as they pass through the scene, longer trajectories are a priori more likely to belong to the static background of the scene [21]. Furthermore, small trajectories are more likely to be due to errors related to bad feature point tracking [8, 62]. The spatial distribution can also be exploited over a period of time. By using

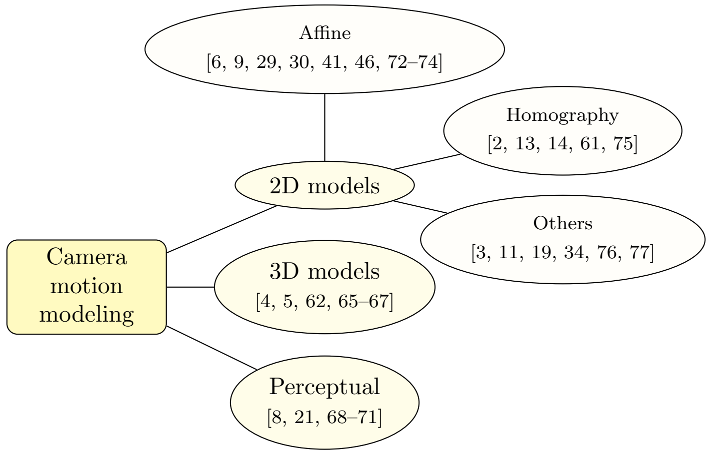

Figure 5: Main approaches for camera motion modeling

clustering techniques, the scene can be divided into background/foreground, the moving objects are then removed by suppressing clusters with greater compacity [8].

3.3. Camera motion modeling

Once the outliers have been removed, the remaining movements are the result of the camera motion. They can therefore be used to model or approximate the camera motion. To that end, two strategies can be used. In most works, the modeling of the camera motion is based on geometrical models that describe the physical process of capturing a scene with a pinhole camera. Early works have proposed to use 2D models that approximate the effects of camera motion on the movements of pixels in the video (Section 3.3.1). By recovering the depth information and by using 3D models, it is also possible to derive the original 3D displacements of the camera (Section 3.3.2). Alternatively, another approach is to avoid geometrical models in order to obtain perceptually plausible models with visually acceptable corrections rather than physically accurate ones (Section 3.3.3).

3.3.1. 2D models

As such, the physical movement of the camera lies in a 3D space. However, the influence of the camera movements is only accessible through the frames of the video, i.e. a 2D space. This is why 2D models approaches do not attempt to recover the original 3D path of the camera but model its influence between two frames as a 2D transformation. More specifically, considering two successive frames I_t and I_{t+1}, and a pixel $\mathbf{z}_t$ belonging to frame t, its coordinates $\mathbf{z}_{t+1}$ in frame $t+1$ are given by

$$\mathbf{z}_{t+1} = H_t \mathbf{z}_t \tag{1}$$

where H_t is a 2D-transformation matrix describing the motion between frames t and $t+1$. The general form of the H_t matrix is expressed as follows,

$$H_t = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}, \tag{2}$$

allows to consider several types of 2D transformations such as pure translation, pure rotation, similarity, affinity or homography. 2D approaches have been really popular for their simplicity of use and low computational cost [19]. They do not require the challenging task of depth estimation, and provide, in case of low parallax or small relative depth variations, a fast and robust way of determining camera movements [30, 78]. Moreover, the 2D assumption is often valid on a local temporal scale when the movements of the camera are not too large or too abrupt.

The simplest parametric planar transforms to be considered are the similarities (also referred to as simplified affine models) [19]. They are able to handle translations, scaling and rotation along the camera axis [34]. The estimation of the four parameters can be solved by linear Least Squares Method on a set of redundant equations [39, 79, 80], possibly combined with filtering/outlier removal [33, 44]. Histogram-based approaches have also proven to be effective in this context [76, 81]. Empirical studies have shown that, in videos acquired with hand-held cameras, most of the involuntary movements such as vibrations are considered significant in the plane perpendicular to the z-axis [78]. These results show that by considering only scale, z-axis rotation, and translations, it is possible to obtain an acceptable approximation [11]. Indeed, the impact of pitch and yaw rotations on the final image warping are often low for this kind of videos. Furthermore, due to its low number of parameters to be estimated, the similarity model constitutes a relevant solution for real-time applications [3]. The similarity model also presents the advantage of introducing very low deformations, which makes it robust to outliers and noise [77]. However, it cannot account for strong rotations outside the camera axis , which may limit its performances in the case of strongly degraded videos.

Slightly more complex with six parameters, the affinity (or generalized affine model) is the most commonly used 2D model. The affinity model encompasses most of the advantages of the similarity model, but additionally allows the possibility to consider also shear [6, 38, 40, 47]. The parameters can be estimated with standard differential motion techniques [29], with more complex cost functions [30] or through multi-scale [72] or hierarchical analysis [32, 73]. The major advantage of using an affinity model lies in the fact that it naturally handles global motions, for which the affinity parameters at every location are assumed to be the same [74]. The model can deal with scenes containing small relative depth variations and zooming effects [41] and provides an acceptable compromise between accuracy and computational cost[46]. However, being a 2D planar transform, it cannot model non-linear inter-frame motion [9].

Finally, the most exhaustive 2D geometrical transform is the homography, which uses 8 parameters. While the interpretation of the coefficients is not as straightforward as for the similarity or the affinity transform [2], they control rotations, translations, zooming and sheering in the x- and y- axis [75]. Homographies have been widely used for image registration purpose [61], and most authors use techniques developed in this context for the estimation of the eight parameters [14, 16, 38]. Nevertheless, the homography model has the potential to cause severe deformations, particularly in the presence of outliers [13].

Other 2D models include simpler models with 3 [35, 77, 82] or 4 parameters (2 rotations and 2 translations) [83]. So-called 2.5D models represent a compromise between 2D and 3D models, by considering cases where 3D displacements can be simplified to avoid the need for depth parameter (translations 1 axis (x, y or z)) [36].

3.3.2. 3D models

In contrast to 2D models, 3D models aim at recovering the actual original 3D displacement of the camera, which is represented by a single point, according to the standard pinhole camera assumption. Instead of only considering the influence of the camera motion in the 2D plane, the 3D models aim to provide an alternative framework that allow to represent realistic displacements in all directions. It is worth to noticing that the motion estimation accuracy strongly depends on the depth recovery step [62, 84]. Recovering depth consists in analyzing the 2D information on the original video, in order to retrieve the original 3D content of the scene. This task, referred to as structure-from-motion [20], often uses groups of 3 key-frames and estimates the parametric 3D transformation that best fit the observed movement. In practice, the computation of the model may be subject to numerical instabilities, especially if the movement is not strong enough. To that end, it is common to use distant key frames, so as to capture sufficient motion information. However, if the motion contains no depth differences and/or no translations, numerical instabilities are inevitable. To handle this issue, some recent works propose to add geometrical constraints in the model (existence of planes [66], manifold constraints [65, 85]) to ensure good accuracy. The computation cost, which is often high, can be kept under control by focusing only on particular regions of interest [67, 86].

The main drawbacks of 3D models can also be addressed by building hybrid models that combine 2D models (which are efficient, easy to compute but imprecise) and 3D models (which are physically accurate but tricky to compute). To that end, some methods propose to consider only certain displacements in the 3D space. For instance, by considering only rotations, it is possible to drop the depth recovery task [4, 7]. This assumption appears to be valid for hand-held shakes but is violated in more complex scene context such as walking or driving. In the context of moving vehicles, plausible movements are limited to rotations and translation in the direction of the car displacement [87]. By using these constraints, it is possible to simplify the general 3D model by using ad-hoc methods less complex than structure-from-motion approaches [5].

3.3.3. Perceptual models

As seen in the previous subsections, several motion models have been proposed in the literature. Selecting the appropriate model can be tricky when no extra information is available on the camera movement, which is, unfortunately, often the case. Moreover, the models perform very differently depending on the scene and the camera motion, and choosing an inappropriate model can have severe repercussions on the stabilization results [21]. As a result, some recent methods prefer to avoid geometrical models and instead use models that provide visually plausible videos rather than physically accurate ones.

One way to distance from geometrical models is to relax the constraints of geometrical plausibility and use the 2D/3D models as ad-hoc local transformations. For instance, homographies are known to provide a good approximation of the camera motion in many situations, and mainly fail in the presence of parallax. In order to combine the robustness of this model and avoid the distortions caused by depth differences, the camera can be modeled by a mixture of homographies [71]. An

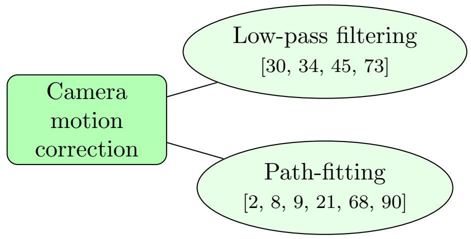

Figure 6: Main approaches for camera motion correction

initial global homography is used to fit one frame onto the next, and localized homographies are charged with describing the residual motion from the global homography. This approach provides more flexibility while remaining robust, but does not correspond to a physical model.

Another way to avoid a specific model is to exploit a known property of camera movements: the movements resulting from the camera motion can be approximated over a small time window by a low rank subspace method [88]. By projecting the trajectories recovered from feature points into a low-rank subspace through matrix factorization techniques, it is possible to extract eigen-trajectories that model the dominant movements in the video [21]. The eigen-trajectories can then be smoothed like any other motion model parameters to derive the smoothed trajectories. A moving SVD-based factorization strategy is used to project the entire trajectory matrix into a 9-rank subspace. Tang et al.[89] use the same principle but with alternative factorization methods either based on a sparse representation strategy to improve the factorization , or using local projection rather than projecting all trajectories in the same subspace [69].

3.4. Camera motion correction

Once the camera motion has been modeled, new camera movements should be determined in order to improve the video synthesis quality. This begs the question : what types of camera motion should be used as input for the synthesis process? Since one of the most problematic aspects of camera motion is the high-frequency shakes, which cause considerable visual discomfort. One solution is to use filtering to remove such annoying distortions (Section 3.4.1). Another possibility is to look to cinematographic considerations for the type of motion used (Section 3.4.2).

3.4.1. Filtering

Temporal filters can be used to remove unwanted components of camera motion. Depending on the camera motion model, they can be applied to feature trajectories to derive the stabilized position of feature points, or to the camera motion parameters to obtain its stabilized position. In the latter case, parameters are generally treated separately. A very common filter is the Gaussian filter [7, 32, 41, 46, 66, 70, 73–75]. It suppresses the high-frequency motion that are the most detrimental to visual comfort. It also offers the possibility to adjust the level of stabilization by the width by tuning the Gaussian kernel. Another common solution is the Kalman filter [5, 16, 30, 33, 47]. It uses the observed motion to estimate the intentional motion and the unwanted motion to be corrected. Motion Vector Integration [34, 43] combines the current and previous frame-to-frame motion to determine a stable camera motion with the initial frame as reference. Finally other methods use second order filters [36, 45] with cut-off frequencies based on the considered applications.

3.4.2. Path-fitting

Determining objectively what could be considered as "good" path for the camera motion is rather difficult question to answer. Several criteria can be taken into consideration to address this issue. The most important criteria are motion regularity or smoothness and resolution loss. One approach is to fit a particular model to the camera motion, in particular polynomial models. Constant models simulate still shots, linear models simulate tracking shots, and quadratic models can simulate the transition from one to other. This has been combined with user-input to select desired motion type [84, 91]. However, these models do not hold for long video sequences. Therefore, the quadratic models can be replaced by spline interpolation, with control point chosen by the user [21]. To avoid user-input while handling longer sequences, fitting quadratic models over time-windows rather than the entire sequence and combining them with a Gaussian filter has also been proposed [41]. Another similar method is Re-cinematography [2], which automatically detects static segments in the camera motion and use tracking shots to link different static segments. To avoid sudden accelerations, quadratic motion is used at the junction between static and tracking segments in order to generate a smooth transition. Another approach is to use an energy minimization scheme to determine the new camera motion [8, 37, 71]. The regularity of the camera motion is represented by one or more energy terms depending on the desired camera motion. The loss of resolution is either used as a hard constraint on the smoothed camera path or as another energy term. Other

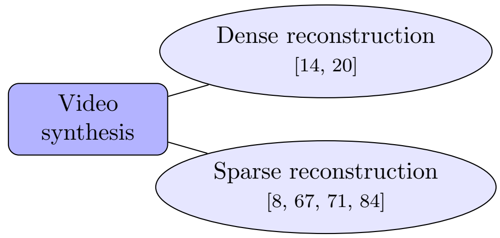

Figure 7: Main approaches for video synthesis

considerations are easy to implement as additional energy terms or constraints. The first term is often the deviation from the original path that is often combined with several order derivatives. L1 regularizations can be used to minimize the first derivative [3, 8, 13, 31, 65, 68, 79], the second derivative [8, 9, 90] or the third derivative [9]. L2 regularizations have also been used to minimize the second derivative [44].

Additional constraints can be used to avoid distortion artifacts, with constraints on spatial rigidity [37]. Weights are typically used to balance the data and regularity terms, and can be adapted to preserve motion-discontinuity which, while visually distracting, would cause heavy resolution loss to recover during the rendering step [71].

3.5. Video synthesis

Once the rectified camera path has been computed, a new video corresponding to this path is generated. This step depends on the choice of the camera model. Most geometrical models describe the original and corrected motion for each pixel, allowing for dense reconstruction where the all modifications applied are known (Section 3.5.1). However, some models only describe the motion, both original and rectified, for certain points. A sparse reconstruction is then used to spread known corrections to all pixels (Section 3.5.2). Two types of reconstruction, described in the following, are reported in the literature.

3.5.1. Dense reconstruction

In the case of dense reconstruction, the original and stabilized positions are either known or computed for every pixel in any given frame. This is the case for approaches using 2D or 3D geometrical motion models or for approaches based on dense optical flow.

Methods using transformation matrices aim at finding the transformations that map the original motion to the corrected one. In order to apply the smoothed transform back to the original video, it is necessary to define a reference frame on which all other frames will be aligned. In this case, one frame, usually the first, can serve as a global reference to the rest of the video. Compositional methods multiply the different transformations between the reference frame and the considered frame; while additive methods compute the cumulative parameters between the reference and the considered frame. Alternatively, in additive methods, each frame can serve as a local reference to compute its the corresponding rectified motion. This allows to reduce the errors accumulation when estimating the motion parameters from distant frames [14]. The stabilized frame is then obtained by applying the transformation to each coordinate of the frame. In the case of 3D transformations that take translation into account, depth maps are recovered using structure-from-motion method (SFM) [20]. In the case of perceptual models using dense flow fields, the original optical flow is mapped to the stabilized optical flow.

3.5.2. Sparse reconstruction

Sparse reconstruction is needed when the motion correction is known for only certain pixels. It seeks to spread the corrections known at certain points to the rest of the image. It is necessary when using 3D models with a sparse depth map or using single-frame warping, as well as perceptual models that focus on stabilizing a select number of trajectories. Content-preserving warps (CPW) [84] spreads the correction from known pixels to the rest of the frame by applying a 4×4 grid over the video frame over the image, and using the known corrections to warp the grid, and the frame with it. The position of each pixel and feature point is expressed as a function of the enclosing quad vertices, so that changing the position of the vertices also alters the position of the enclosed pixels. New vertices positions are computed using an energy minimization method, with two components that control the trade-off between stabilization of the trajectories and the preservation of structures in the video. Once the new vertices are known, the corrected position of each pixel is obtained using the initially computed weights and the new vertices positions. This warping scheme is used in several stabilization methods [67, 71], and has served as the basis for other warping schemes. For instance, a tighter mesh (with each quad approximately 10x10 pixels) is used along with reliability weights for each feature points, to avoid the influence of outliers [37]. The similarity constraint is also changed to a homography constraint. Finally, Koh et al. [8] add a regularity constraint based on the distance between the centers of adjacent quads, and change the structure constraint to maintain right angles for each quad.

4. Performances and evaluation

This section presents an overview of the performance aspects of video stabilization. First, we present the current situation in terms of performances. The second part is dedicated to the evaluation tools used for Video Stabilization Quality Assessment (VSQA).

4.1. Current performances of video stabilization methods

Despite the many published studies on VS algorithms, there is currently no consensus on which approach is the most effective in real applications. It is worth noticing that even recent methods do not clearly outperform the standard approaches developed in the early 2010s [37, 92–95]. Commercial solutions such as VirtualDub Deshaker, Google Youtube Stabilizer or Adobe After Effects, based on concepts introduced in the late 2000s/early 2010s, are still considered as valid and competitive state-of-the-art methods [96]. In particular, they offer the nice property to be more robust than several recent methods and thus achieve a good compromise between signal stabilization and degradation caused by uncontrollable side effect [97].

All recent methods can easily handle simple scenarios corresponding to low-complexity scenes where the movement amplitude is relatively low. In the context of crowd scene with several objects in movements, the performances decrease irrespective of the used stabilization method. For particular cases such as body-worn, vehicles or traffic camera, performances can be improved by adapting the models to the considered task [87].

Furthermore, to provide a general methodology and guidelines on how to assess the performance of VS methods in different scenarios and contexts remain rather a difficult task. Indeed, although significant effort has been spent over more than two decades to develop efficient VS methods, there is currently no well defined and accepted unifying framework for evaluation of the VS results. The first obstacle is linked to the difficulty of building ground truths for the stabilization task. Most VS methods conceived until 2018 where only tested either on synthetic data or real data without ground truths. Several evaluation databases have been introduced in the early literature are refined over the years [8, 9, 21, 58, 71, 84], but up to our knowledge the first significant database composed of pairs or stable/unstable videos was published in 2018 [96]. Since then, several large-scale databases have been published [98, 99], but it a still a promising and poorly studied research topic. Table 2 displays the characteristics of the main databases published online.

4.2. Video Stabilization Quality Assessment

Video Stabilization Quality Assessment (VSQA) is the process of evaluating the performance of the VS algorithms in terms of perceptual quality or pleasantness. It could be performed using different methods and protocols based on some criteria and statistical analysis tools. It could be based on subjective evaluation or objective measurement. It is inherently a multi-criterion problem since the visual discomfort due to the camera motion, artifacts caused by the stabilization process such as resolution loss and distortions all contribute to the quality of the output video. All these artifacts and distortions are unfortunately not easy to handle in a mathematically tractable model.

To the best of our knowledge the work done by Balakirsky and Chellappa [100] was the first published study on performance evaluation of VS algorithms. In this pioneer study only three VS algorithms were evaluated using a limited set of objective criteria restricted to a dedicated tracking application. This work has been revised and augmented in [101]. Since these two pioneer works, a few studies have been reported in the literature [93, 96, 97, 102–106]. However, there is no satisfying methodology to evaluate the perceptual quality of the VS outputs in effective way through a well defined and acknowledged unifying framework. Probably for these reasons, although many video stabilization methods have been proposed, a little attention has been paid to video stabilization quality assessment. Despite the already mentioned few interesting studies, VSQA therefore appears as a hot and very challenging research topic. Most of the proposed objective measures for VSQA are based on some heuristics and intuitive approaches. In the literature, two main approaches have been used to assess the results of video stabilization methods. Very often the quality of the processed video is evaluated simply by visual inspection or with user studies on a panel of observers (Section 4.2.1). Objective metrics have also been introduced but are often limited to basic quantitative measures such as inter-frame PSNR-based metrics (Section 4.2.2)or other simple geometrical distortion measures [14, 42, 69].

4.2.1. Subjective evaluation

The main objective of video stabilization is to improve the visual comfort of viewers, which is inherently a subjective goal. Indeed, the video stabilization quality is highly related to several aspects, such as the choice of the substituted/virtual camera path or the correction of rolling shutter, that depend on psycho-visual criteria that are not completely understood. For these reasons, subjective evaluation is often preferred to objective metrics.

Despite the great importance of the quality aspect, as the video is supposed to be viewed by human observer, to the best of our knowledge, apart very few studies, most published work do not provide a throughout subjective evaluation of the VS outputs. The VSQA is very often neglected or left to the subjective evaluation of readers.

At the present there is no unified benchmark framework to carry out the formal comparison of the VS methods. This is mainly due to the complexity of the process by which the observers evaluate the outputs. To the best of our knowledge, only four comprehensive studies, where VSQA aspects are taken into account, have been reported in the literature [6, 8, 70, 71]. These studies are based on pairwise comparisons between different methods. Subjects are shown two juxtaposed videos and are asked to select the best one, according to some predefined criteria. For instance, in [8], users were instructed to neglect differences in aspect ratios, contrast or sharpness and to focus on deformations of scene structures, rolling shutter distortions, and wobbling or shaking. In [6] the authors used four criteria: the stability of the video content, whether any distracting objects were present, the quality of the movements of the camera, and how much of the scene was removed by cropping. On the contrary, users in [71] were told to ignore differences in ratio or sharpness, which could be caused by the codec used in the algorithms. In Wang et al.approach [70] each participant was asked to give to each stabilization result a score between 0 and 100 without providing any analysis criterion or guideline. Interestingly, each of these protocols is different. There is no consensus on the different steps of the process (rating system, presence of the original video, possibility to have no preference, etc.). In particular, instructions given to the users vary widely, which may have an impact on the outcome of the study since they were asked to focus on only a few aspects of video stabilization. Finally, all these studies aimed at demonstrating the efficiency of one algorithm over the state-of-the-art methods: there might therefore exist a bias in the experiment design that voluntary focused on the positive aspects of the proposed algorithm.

Besides the already mentioned limitations, user-studies are also time-consuming and difficult to set up. This is probably the main reason why very few studies were reported in the literature. However, there is still a need for subjective evaluation since it is the most reliable and complete way of assessing video stabilization quality [107].

4.2.2. Objective evaluation

Several criteria have been used to design objective VSQA measurements. Some of the video stabilization components could be evaluated using objective metrics. For instance, the accuracy of the original camera path estimation can be evaluated when the ground truth or a priori knowledge on the original path are available. While ground truth is usually unavailable in real scenarios, it could be built for synthetic video sequences. The fitting error between the ground-truth position of features and those resulting from the estimated movements could be used as an objective evaluation of the camera motion estimation [8, 38, 69]. Using real videos taken so that the camera is in the same position at the start and end of the sequence, the differences in camera pitch can similarly measure the estimation accuracy [4].

Another criterion is to use peak signal to noise ratio (PSNR) or the structural similarity (SSIM) to measure dis-alignment between two successive frames. However, this is only reliable when the stabilization process removes or dampens the camera motion, and does not allow for intentional movements [13]. In addition, the PSNR is sensitive to illumination changes and moving objects, and is strongly influenced by contrast variation. The inter-frame transformation fidelity (ITF) which measures the average inter-frame PSNR [76, 78], has also been used. Closely related is the contrast-invariant error measure [7], which uses the normalized differences in colour over small patches. This normalization avoids capturing differences in uniform areas that are not sensitive to small camera shifts.

Measuring artifacts, resolution loss or other distortion for a fixed degree of stabilization is another possibility. For instance a distortion measure based on the eigenvalues of the correcting transformations was proposed in [71]. The use of the percentage of undefined area between the motion correction and cropping to measure resolution loss[14, 42, 69] is another alternative for quantifying the VS quality. Using those metrics implicitly introduce somewhat a bias in measuring the level of VS quality. Indeed, these metrics focus only on a single step of the stabilization process used to evaluate 2-d motion models and composition strategies [14]. Recently a no-reference VSQA metric has been proposed [97]. It is mainly based on the analysis of the intrinsic smoothness of the motion path modeled through a mathematical formalism based on a high-dimensional manifold of Lie group theory. The same authors pointed out the limitations of the proposed metrics. Indeed, all the considered metrics consider only one aspect of VS (optical flow, smoothness of motion, VS side effects, inter-frame similarity, etc) but do not consider any aspects on the whole appearance of the video as annoying level and global visual discomfort. Since these metrics focused on limited aspects of the VS pipeline they would be inevitably in favor of the method incorporating such aspects. The VS evaluation based on this kind of metrics would be then quite biased and useless. Furthermore, these quantitative measures do not exploit any a priori knowledge on the distortions nor well-defined model of the visual discomfort due to video instability. These measures only assess a small subsets of the features or characteristics of motion that are considered as the main origins of such annoying distortion.

5. Challenges and perspectives

As reported in this contribution, video stabilization has seen considerable progress in recent years. However, many challenges and open problems remain, in particular regarding the evaluation of the stabilization process.

5.1. Current challenges

Lack of evaluation framework. Despite several propositions, there is still no accepted metric to quantify the quality of video stabilization. Indeed, as far as we know there is no reference data-set to test different stabilization processes, as different scenes lead to very different challenging issues. This is particularly problematic, as video stabilization necessitates a trade-off between the removal of unwanted motion and the loss of resolution, video perceptual quality and other side-effects that may result from the different processing of the input signal. Without an effective objective measure, this trade-off needs to be fixed heuristically.

Running time and real-time configurations. Other difficulties involve the running time: videos can take a long time to process, and striving for real-time stabilization requires not only a computationally very efficient process but will also lack information about the future camera movements, complicating the computation of the stabilized camera path. Recent approaches have proposed to handle this issue by conceiving fast algorithms allowing to process up to 30 frames per second, hence allowing a quasi real-time processing [108, 109]. These algorithms are suitable for a use on smartphones, with a reduced latency down to 1 frame [99, 110], while still being purely based on software, i.e. digital video stabilization. Other popular solutions embedded on smartphones are related to Optical Image Stabilization hardware [111], which often require the use of additional sensors such as gyrometers, that allow for a precise and real-time stabilization [112] for both photos and videos. We refer the readers to several interesting reviews on the topic [112, 113] for more information.

Moving objects/subjects Another challenging problem is the presence of moving objects. Moving objects often result in occlusions, which are challenging and can degrade the quality of motion detection. Their presence can also mislead the camera motion estimation, particularly large moving objects, which are frequently mistaken for the dominant camera movement. The detection of moving objects, while remaining difficult, has seen promising results by exploiting either motion discontinuities or the evolution of motion over time.

Models and parameters The choice of motion model is critical, as 3D models are computationally demanding and unstable, while 2D models are insufficient to model scenes containing strong parallax. Perceptual models meanwhile strive for a middle ground, but at the risk of deformations and physically-inaccurate results. While full 3D models remain unstable, 2D models have proven efficient for small degrees of stabilization or for scenes lacking parallax, while perceptual models have shown a wider range of stabilization without excessive loss of robustness.

5.2. The future of video stabilization: deep learning approaches and beyond

Like all other research topics VS could not escape the breaking wave raised by Deep Learning (DL) approaches. According to the authors of [99], StabNet method was the first online VS based learning approach published in the literature. Since then, several works have shown that the use of DL approaches seems to be a promising research direction for VS [94, 98, 99, 114]. In their pioneer work, Wang et al. [99] perform VS by using an encoder/multi-grid-regressor architecture that uses sequences of images as inputs. The loss function includes three terms, namely stability, time and shape-preserving constraint inspired from [84]. Similarly to classical DL methods, the stabilization process is fast, but the training of the network requires a large amount of annotated data. Xu et al. [114] build on these principles and use a transform-aware encoder/decoder structure with several networks instead of one. However, their method only output a global 2D affine model between two adjacent frames, which can be tricky for complex videos. In order to address these issues, two main directions have emerged in the literature and may constitute the future of DL-based VS. The first one is related to the difficulty of generating massive training data, that can be avoided by using unsupervised approaches. For instance, Yu et al. [95] learn the adequate 2D model through a CNN network, that is only used as a regression tool, thus removing the training step. However, this

CNN-optimizer based VS method is computational very demanding as mentioned by the authors. Choi et al. [115] also propose a unsupervised approach, that focuses on the problem of image interpolation that is crucial in VS due to the cropping effect. They outperform standard non-DL approaches, especially according to the cropping ratio metric. However, the process may introduce blur in the stabilized videos. The second current direction for DL-based VS is to remove (or at least relax) the use of 2D model such as affine transforms of homographies. To that end, and similarly to what happened for classical VS, authors now attempt to replace the fixed models by perceptual motion models. Zhao et al. [94] introduce PWStableNet, to learn pixel-wise warping maps, which can be seen as a flexible model that generalizes homographies. This technique called PWStableNet has been proven more robust and computationally efficient than some representative state-of-the-art methods including DL-based approaches [99, 114] and other traditional techniques such as Adobe Premiere Video Stabilizer, but as all supervised DL techniques, their method is inherently dependant on the training database.

From this brief overview of CNN-based methods, our feeling is that the contribution of approaches that are directly or indirectly based on the current trend, i.e. deep learning, is undoubtedly promising and that it requires more hindsight and effort for better exploitation, and that it is not appropriate to rush to follow the DL wave without measuring its limits and strengths.

An important aspect that needs to be considered, and especially when considering DL approach, is the lack of a sufficient number of datasets dedicated to perceptive VS evaluation as is the case for classical image quality assessment. To the best of our knowledge only two datasets with the psycho-visual test results are available [97, 98]. The other issue is related to the necessity of building efficient VSQA metrics, that could be used as loss functions in the DL optimization scheme. Current DL methods often distinguish three main aspects of VS: cropping ratio (remaining area after cropping away empty regions), distortion value (degree of distortion of stabilization results compared to original ones) and the stability score (smoothness of the stabilized videos). These three terms are often used as loss functions for the training of the networks and also for evaluation: it is therefore not surprising to see these metrics in favour of the authors' method as they are very much related to the concepts and ideas of the proposed video stabilization schemes. However, none of these metrics actually reflect the perceptive aspects of VS.

We believe that the future of VS may rely on the fusion between machine learning approaches such as DL and the use of reviews and feedback from users. This is not only motivated by the fact that we are witnessing a deep revolution in computer vision world, but we are convinced that introducing the learning process in the VS pipeline would allow to take into account the human perception and as a result yields to more efficient video stabilization solutions. For instance, reinforcement learning techniques could be useful in order to really base the stabilization methods on subjective evaluation rather than geometrical criteria that do not take into account all perceptual aspects of VS. We believe that the DL based approaches will increasingly emerge as a potential solution for video stabilization in the coming years. Furthermore, we believe that DL based approaches could be better improved by incorporating perceptual approaches for visual information processing [116] and particularly perceptual models of motion perception and analysis [117, 118].

References

[1] Daniel Durini. *High performance silicon imaging: fundamentals and applications of CMOS and CCD sensors*. Elsevier, 2014.

[2] Michael L Gleicher and Feng Liu. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the ACM International Conference on Multimedia*, pages 27–36, 2007.

[3] Hung-Chang Chang, Shang-Hong Lai, and Kuang-Rong Lu. A robust real-time video stabilization algorithm. *Journal of Visual Communication and Image Representation*, 17(3): 659–673, 2006.

[4] Chao Jia and Brian L Evans. Probabilistic 3-D motion estimation for rolling shutter video rectification from visual and inertial measurements. In *Proceedings of the IEEE International Workshop on Multimedia Signal Processing (MMSP)*, pages 203–208, 2012.

[5] Yu-Ming Liang, Hsiao-Rong Tyan, Shyang-Lih Chang, H. Y. M. Liao, and Sei-Wang Chen. Video stabilization for a camcorder mounted on a moving vehicle. *IEEE Transactions on Vehicular Technology*, 53(6):1636–1648, 2004. doi: 10.1109/TVT.2004.836923.

[6] Fang-Lue Zhang, Jue Wang, Han Zhao, Ralph R Martin, and Shi-Min Hu. Simultaneous camera path optimization and distraction removal for improving amateur video. *IEEE Transactions on Image Processing*, 24(12):5982–5994, 2015.

[7] Erik Ringaby and Per-Erik Forssén. Efficient video rectification and stabilisation for cell-phones. *International Journal of Computer Vision*, 96(3):335–352, 2012.

[8] Yeong Jun Koh, Chulwoo Lee, and Chang-Su Kim. Video stabilization based on feature trajectory augmentation and selection and robust mesh grid warping. *IEEE Transactions on Image Processing*, 24(12):5260–5273, 2015.

[9] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust L1 optimal camera paths. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 225–232, 2011.

[10] David Sachs, Steven Nasiri, and Daniel Goehl. Image stabilization technology overview. *InvenSense Whitepaper*, 2006.

[11] Qinghe Zheng and Mingqiang Yang. A video stabilization method based on inter-frame image matching score. *Global Journal of Computer Science and Technology*, 2017. ISSN 0975-4172.

[12] H. N. Thi Quynh, Trung Dang, Beghdadi Azeddine, Ammar Heyfa, and Benazza Amel. Automatic blotch removal using a perceptual approach. In *Proceedings of The 11th International Conference on Knowledge and Systems Engineering (KSE 2019)*, 2019.

[13] Semi Jeon, Inhye Yoon, Jinbeum Jang, Seungji Yang, Jisung Kim, and Joonki Paik. Robust video stabilization using particle keypoint update and L1-optimized camera path. *Sensors*, 17(2):337, 2017.

[14] Javier Sánchez. Comparison of motion smoothing strategies for video stabilization using parametric models. *Image Processing Online*, 7:309–346, 2017.

[15] Andrea Teatini, Congcong Wang, Rafael Palomar, Faouzi Alaya Cheikh, Azeddine Beghdadi, Bjørn Edwin, and Ole Jakob Elle. Validation of stereo vision based liver surface reconstruction for image guided surgery. In *Colour and Visual Computing Symposium (CVCS)*, pages 1–6, Gjøvik, Norway, 2018. doi: 10.1109/CVCS.2018.8496589. URL `https://doi.org/10.1109/CVCS.2018.8496589`.

[16] Jing Dong and Haibo Liu. Video stabilization for strict real-time applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):716–724, 2017. doi: 10.1109/TCSVT.2016.2589860.

[17] Philippe Loic Marie Bouttefroy, Abdesselam Bouzerdoum, Son Lam Phung, and Azeddine Beghdadi. Integrating the projective transform with particle filtering for visual tracking. *EURASIP J. Image and Video Processing*, 2011, 2011. doi: 10.1155/2011/839412. URL `https://doi.org/10.1155/2011/839412`.

[18] Feng Liu, Yuzhen Niu, and Hailin Jin. Joint subspace stabilization for stereoscopic video. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 73–80, 2013.

[19] Carlos Morimoto and Rama Chellappa. Fast electronic digital image stabilization. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, volume 3, pages 284–288, 1996.

[20] Guofeng Zhang, Xueying Qin, Wei Hua, Tien-Tsin Wong, Pheng-Ann Heng, and Hujun Bao. Robust metric reconstruction from challenging video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2007.

[21] Feng Liu, Michael Gleicher, Jue Wang, Hailin Jin, and Aseem Agarwala. Subspace video stabilization. *ACM Transactions on Graphics (TOG)*, 30(1):4:1–4:10, 2011.

[22] Toshio Moriya, Fumiko Shiojiri, and Haruo Takeda. Dynamic 3d stabilization for video CG composite. In *Proceedings Fourth IEEE Workshop on Applications of Computer Vision, WACV 1998, October 19-21, 1998, Princeton, New Jersey, USA*, pages 260–261, 1998. doi: 10.1109/ACV.1998.732897. URL `https://doi.org/10.1109/ACV.1998.732897`.

[23] Matthias Grundmann, Vivek Kwatra, Daniel Castro, and Irfan A. Essa. Calibration-free rolling shutter removal. In *Proceedings of the IEEE International Conference on Computational Photography (ICCP)*, pages 1–8, Seattle, WA, USA, 2012. doi: 10.1109/ICCPhot.2012.6215213. URL `https://doi.org/10.1109/ICCPhot.2012.6215213`.

[24] Zhongqiang Wang, Lei Zhang, and Hua Huang. High-quality real-time video stabilization using trajectory smoothing and mesh-based warping. *IEEE Access*, 6:25157–25166, 2018. doi: 10.1109/ACCESS.2018.2828653. URL `https://doi.org/10.1109/ACCESS.2018.2828653`.

[25] Chao Zhang, Fugen Zhou, Bindang Xue, and Wenfang Xue. Stabilization of atmospheric turbulence-distorted video containing moving objects using the monogenic signal. *Sig. Proc.: Image Comm.*, 63:19–29, 2018. doi: 10.1016/j.image.2018.01.006. URL `https://doi.org/10.1016/j.image.2018.01.006`.

[26] Jiyang Yu and Ravi Ramamoorthi. Selfie video stabilization. In *Proceedings of the IEEE European Conference on Computer Vision (ECCV)*, pages 569–584, Munich, Germany, 2018. doi: 10.1007/978-3-030-01228-1_34. URL `https://doi.org/10.1007/978-3-030-01228-1\_34`.

[27] Wilbert G. Aguilar, David Loza, Luis Segura, Alexander Ibarra, Thomas Abaroa, and Ronnie Fuertes. Onboard video stabilization for rotorcrafts. In *Proceedings of the International Conference on Intelligent Robotics and Applications (ICIRA)*, pages 695–702, Wuhan, China, 2017. doi: 10.1007/978-3-319-65292-4_60. URL `https://doi.org/10.1007/978-3-319-65292-4\_60`.

[28] Sibren van Vliet, André Sobiecki, and Alexandru Telea. Joint brightness and tone stabilization of capsule endoscopy videos. In *Proceedings of the International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*, pages 101–112, Funchal, Madeira, Portugal, 2018. doi: 10.5220/0006552401010112. URL `https://doi.org/10.5220/0006552401010112`.

[29] Hany Farid and Jeffrey B Woodward. Video stabilization and enhancement. Technical report, TR2007-605, Dartmouth College, Computer Science, 1997.

[30] Andrey Litvin, Janusz Konrad, and William C Karl. Probabilistic video stabilization using kalman filtering and mosaicing. In *Image and Video Communications and Processing*, pages 663–674, 2003.

[31] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Steadyflow: Spatially smooth optical flow for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4209–4216, 2014.

[32] Yasuyuki Matsushita, Eyal Ofek, Xiaoou Tang, and Heung-Yeung Shum. Full-frame video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 50–57, 2005.

[33] NA Tsoligkas, D Xu, I French, and Y Luo. A motion model based video stabilisation algorithm. In *Proceedings of the World Automation Congress (WAC)*, pages 1–6, 2006.

[34] Sebastiano Battiato, Giovanni Puglisi, and AR Bruna. A robust video stabilization system by adaptive motion vectors filtering. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 373–376, 2008.

[35] J. Narendra Babu, M. Nageswariah, S. Shajahan, and A. Maheswari. Block processing video stabilization. *International Journal of Scientific and Research Publications (IJSRP)*, 3(3), 2013.

[36] Zhigang Zhu, Guangyou Xu, Yudong Yang, and Jesse S Jin. Camera stabilization based on 2.5 D motion estimation and inertial motion filtering. In *Proceedings of the IEEE International Conference on Intelligent Vehicles*, pages 329–334, 1998.

[37] Yu-Shuen Wang, Feng Liu, Pu-Sheng Hsu, and Tong-Yee Lee. Spatially and temporally optimized video stabilization. *IEEE Transactions on Visualization and Computer Graphics*, 19(8):1354–1361, 2013.

[38] Shiwei Li, Lu Yuan, Jian Sun, and Long Quan. Dual-feature warping-based motion model estimation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4283–4291, 2015.

[39] Sebastiano Battiato, Giovanni Gallo, Giovanni Puglisi, and Salvatore Scellato. SIFT features tracking for video stabilization. In *Proceedings of the International Conference on Image Analysis and Processing (ICIAP)*, pages 825–830, 2007. doi: 10.1109/ICIAP.2007.4362878.

[40] Bing-Yu Chen, Ken-Yi Lee, Wei-Ting Huang, and Jong-Shan Lin. Capturing intention-based full-frame video stabilization. *Computer Graphics Forum*, 27(7):1805–1814, 2008.

[41] Rong Hu, Rongjie Shi, I-fan Shen, and Wenbin Chen. Video stabilization using scale-invariant features. In *Proceedings of the International Conference on Information Visualization (IV)*, pages 871–877, 2007. doi: 10.1109/IV.2007.119.

[42] Ken-Yi Lee, Yung-Yu Chuang, Bing-Yu Chen, and Ming Ouhyoung. Video stabilization using robust feature trajectories. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1397–1404, 2009.

[43] Xie Zheng, Cui Shaohui, Wang Gang, and Li Jinlun. Video stabilization system based on speeded-up robust features. In *Proceedings of the International Industrial Informatics and Computer Engineering Conference (IIICEC)*, pages 1996–1998, 2015.

[44] Chunhe Song, Hai Zhao, Wei Jing, and Yuanguo Bi. Robust video stabilization based on bounded path planning. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 3684–3687, 2012.

[45] Wilbert G Aguilar and Cecilio Angulo. Robust video stabilization based on motion intention for low-cost micro aerial vehicles. In *Proceedings of the IEEE International Multi-Conference on Systems, Signals & Devices (SSD)*, pages 1–6, 2014. doi: 10.1109/SSD.2014.6808863.

[46] Manish Okade and Prabir Kumar Biswas. Video stabilization using maximally stable extremal region features. *Multimedia tools and applications*, 68(3):947–968, 2014.

[47] Jianan Li, Tingfa Xu, and Kun Zhang. Real-time feature-based video stabilization on FPGA. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):907–919, 2017. doi: 10.1109/TCSVT.2016.2515238.

[48] Amar Mitiche and Patrick Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, 1996. doi: 10.1007/BF00131147. URL `https://doi.org/10.1007/BF00131147`.

[49] Auysakul Jutamanee, Xu He, and Pooneeth Vishwanath. A hybrid motion estimation for video stabilization based on an imu sensor. *Sensors*, 18(8):2708, 2018.

[50] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[51] Bruce D Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[52] Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial Intelligence*, 17 (1-3):185–203, 1981.

[53] Deqing Sun, Stefan Roth, and Michael J Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision*, 106(2):115–137, 2014.

[54] Azeddine Beghdadi, Mostefa Mesbah, and Jérôme Monteil. A fast incremental approach for accurate measurement of the displacement field. *Image Vision Comput.*, 21(4):383–399, 2003. doi: 10.1016/S0262-8856(03)00014-3. URL `https://doi.org/10.1016/S0262-8856(03)00014-3`.

[55] Steven S. Beauchemin and John L. Barron. The computation of optical flow. *ACM Comput. Surv.*, 27(3):433–467, 1995.

[56] Denis Fortun, Patrick Bouthemy, and Charles Kervrann. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding*, 134:1–21, 2015.

[57] Carlo Tomasi and Takeo Kanade. *Detection and tracking of point features*. School of Computer Science, Carnegie Mellon Univ. Pittsburgh, 1991.

[58] Amit Goldstein and Raanan Fattal. Video stabilization using epipolar geometry. *ACM Transactions on Graphics (TOG)*, 31(5):126, 2012.

[59] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[60] Sunglok Choi, Taemin Kim, and Wonpil Yu. Robust video stabilization to outlier motion using adaptive ransac. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 1897 – 1902, 2009.

[61] Lionel Moisan, Pierre Moulon, and Pascal Monasse. Automatic homographic registration of a pair of images, with a contrario elimination of outliers. *Image Processing On Line*, 2:56–73, 2012.

[62] Guofeng Zhang, Jiaya Jia, Tien-Tsin Wong, and Hujun Bao. Consistent depth maps recovery from a video sequence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31 (6):974–988, 2009.

[63] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[64] Wilko Guilluy, Laurent Oudre, and Azeddine Beghdadi. Feature trajectories selection for video stabilization. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 593–597. IEEE, 2018.

[65] Chao Jia and Brian L Evans. Constrained 3D rotation smoothing via global manifold regression for video stabilization. *IEEE Transactions on Signal Processing*, 62(13):3293–3304, 2014. doi: 10.1109/TSP.2014.2325795.

[66] Zihan Zhou, Hailin Jin, and Yi Ma. Plane-based content preserving warps for video stabilization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2299–2306, 2013.

[67] Dong-bok Lee, Ick-hyun Choi, Byung Cheol Song, and Tae Hwan Lee. ROI-based video stabilization algorithm for hand-held cameras. In *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 314–318, 2012. doi: 10.1109/ICMEW.2012.60.

[68] K Karageorgos, A Dimou, A Axenopoulos, P Daras, and F Alvarez. Semantic filtering for video stabilization. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6, 2017. doi: 10.1109/AVSS.2017.8078488.

[69] Chengzhou Tang and Ronggang Wang. Local subspace video stabilization. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2014.

[70] Zhongqiang Wang and Hua Huang. Pixel-wise video stabilization. *Multimedia Tools and Applications*, 75(23):15939–15954, 2015.

[71] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM Transactions on Graphics (TOG)*, 32(4):78:1–78:10, 2013.

[72] Michael Hansen, Prabu Anandan, K Dana, G Van der Wal, and Peter Burt. Real-time scene stabilization and mosaic construction. In *Proceedings of the IEEE Workshop on Applications of Computer Vision*, pages 54–62, 1994. doi: 10.1109/ACV.1994.341288.

[73] Paresh Rawat and Jyoti Singhai. Adaptive motion smoothening for video stabilization. *International Journal of Computer Applications*, 72(20), 2013.

[74] Derek Pang, Huizhong Chen, and Sherif Halawa. Efficient video stabilization with dual-tree complex wavelet transform. Technical report, EE368 Project Report, 2010.

[75] Manish Okade and Prabir Kumar Biswas. Fast video stabilization in the compressed domain. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1015–1020, 2012. doi: 10.1109/ICME.2012.113.

[76] Giovanni Puglisi and Sebastiano Battiato. A robust image alignment algorithm for video stabilization purposes. *IEEE Transactions on Circuits and Systems for Video Technology*, 21 (10):1390–1400, 2011. doi: 10.1109/TCSVT.2011.2162689.

[77] Junlan Yang, Dan Schonfeld, Chong Chen, and Magdi Mohamed. Online video stabilization based on particle filters. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 1545–1548, 2006.

[78] Wilbert G Aguilar and Cecilio Angulo. Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 2014(1): 46, 2014.

[79] Hung-Chang Chang, Shang-Hong Lai, and Kuang-Rong Lu. A robust and efficient video stabilization algorithm. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, volume 1, pages 29–32, 2004. doi: 10.1109/ICME.2004.1394117.

[80] Xiaojiang Peng, Junzhou Chen, and Jiashu Zhang. Robust digital image stabilization based on spatial-location-invariant criterion. In *Proceedings of the Annual Conference of the IEEE Industrial Electronics Society (IECON)*, pages 2250–2254, 2011. doi: 10.1109/IECON.2011.6119659.

[81] Giovanni Puglisi and Sebastiano Battiato. Robust video stabilization approach based on a voting strategy. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 629–632, 2011. doi: 10.1109/ICIP.2011.6116630.

[82] Hui Qu and Li Song. Video stabilization with L1-L2 optimization. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pages 29–33, 2013. doi: 10.1109/ICIP.2013.6738007.

[83] Atanas Nikolov and Dimo Dimov. 2D video stabilization for industrial high-speed cameras. *Cybernetics and Information Technologies*, 15(7):23–34, 2015.

[84] Feng Liu, Michael Gleicher, Hailin Jin, and Aseem Agarwala. Content-preserving warps for 3D video stabilization. *ACM Transactions on Graphics (ToG)*, 28(3):44, 2009.

[85] Lei Zhang, Xiao-Quan Chen, Xin-Yi Kong, and Hua Huang. Geodesic video stabilization in transformation space. *IEEE Transactions on Image Processing*, 26(5):2219–2229, 2017.

[86] Tae Hwan Lee, Yun-gu Lee, and Byung Cheol Song. Fast 3D video stabilization using ROI-based warping. *Journal of Visual Communication and Image Representation*, 25(5):943–950, 2014. doi: https://doi.org/10.1016/j.jvcir.2014.02.011. URL `http://www.sciencedirect.com/science/article/pii/S1047320314000492`.

[87] Qiang Ling and Minda Zhao. Stabilization of traffic videos based on both foreground and background feature trajectories. *IEEE Trans. Circuits Syst. Video Techn.*, 29(8):2215–2228, 2019. doi: 10.1109/TCSVT.2018.2862909. URL `https://doi.org/10.1109/TCSVT.2018.2862909`.

[88] Richard I Hartley. Euclidean reconstruction from uncalibrated views. In *Proceedings of the Joint European-US Workshop on Applications of Invariance in Computer Vision*, pages 235–256, 1993.

[89] Chengzhou Tang and Ronggang Wang. Sparse moving factorization for subspace video stabilization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4314–4318, 2014. doi: 10.1109/ICASSP.2014.6854416.

[90] Guofeng Zhang, Zilong Dong, Jiaya Jia, Liang Wan, Tien-Tsin Wong, and Hujun Bao. Refilming with depth-inferred videos. *IEEE Transactions on Visualization and Computer Graphics*, 15(5):828–840, 2009.

[91] Jiamin Bai, Aseem Agarwala, Maneesh Agrawala, and Ravi Ramamoorthi. User-assisted video stabilization. In *Computer Graphics Forum*, volume 33, pages 61–70. Wiley Online Library, 2014.

[92] Marcos R Souza and Helio Pedrini. Digital video stabilization based on adaptive camera trajectory smoothing. *EURASIP Journal on Image and Video Processing*, 2018(1):37, 2018.

[93] Huicong Wu, Liang Xiao, Zhichao Lian, and Hiuk Jae Shim. Locally low-rank regularized video stabilization with motion diversity constraints. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018.

[94] Minda Zhao and Qiang Ling. Pwstablenet: Learning pixel-wise warping maps for video stabilization. *IEEE Trans. Image Processing*, 29:3582–3595, 2020.

[95] Jiyang Yu and Ravi Ramamoorthi. Robust video stabilization by optimization in CNN weight space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3800–3808, 2019.

[96] Lei Zhang, Qing-Zhuo Zheng, Hong-Kang Liu, and Hua Huang. Full-reference stability assessment of digital video stabilization based on riemannian metric. *IEEE Transactions on Image Processing*, 27(12):6051–6063, 2018.

[97] Lei Zhang, Qing-Zhuo Zheng, and Hua Huang. Intrinsic motion stability assessment for video stabilization. *IEEE Trans. Vis. Comput. Graph.*, 25(4):1681–1692, 2019.

[98] Maria Silvia Ito and Ebroul Izquierdo. A dataset and evaluation framework for deep learning based video stabilization systems. In *2019 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4. IEEE, 2019.

[99] Miao Wang, Guo-Ye Yang, Jin-Kun Lin, Song-Hai Zhang, Ariel Shamir, Shao-Ping Lu, and Shi-Min Hu. Deep online video stabilization with multi-grid warping transformation learning. *IEEE Transactions on Image Processing*, 28(5):2283–2292, 2018.

[100] Stephen B. Balakirsky and Rama Chellappa. Performance characterization of image stabilization algorithms. *Real-Time Imaging*, 2(5):297–313, 1996.

[101] Carlos Hitoshi Morimoto and Rama Chellappa. Evaluation of image stabilization algorithms. *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, 5:2789–2792 vol.5, 1998.

[102] Matti Niskanen, Olli Silvén, and Marius Tico. Video stabilization performance assessment. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo, ICME 2006, July 9-12 2006, Toronto, Ontario, Canada*, pages 405–408, 2006. doi: 10.1109/ICME.2006.262522. URL `https://doi.org/10.1109/ICME.2006.262522`.

[103] Marcos Roberto, Helio Pedrini, et al. Digital video stabilization: Algorithms and evaluation. In *Anais do 32º Concurso de Teses e Dissertações*, pages 61–66. SBC, 2019.

[104] Zhaoxiong Cui and Tingting Jiang. No-reference video shakiness quality assessment. In *Asian Conference on Computer Vision*, pages 396–411. Springer, 2016.

[105] Xuelong Li, Qun Guo, and Xiaoqiang Lu. Spatiotemporal statistics for video quality assessment. *IEEE Transactions on Image Processing*, 25(7):3329–3342, 2016.

[106] Wilko Guilluy, Azeddine Beghdadi, and Laurent Oudre. A performance evaluation framework for video stabilization methods. *2018 7th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6, 2018.

[107] Guofeng Zhang, Wei Hua, Xueying Qin, Yuanlong Shao, and Hujun Bao. Video stabilization based on a 3D perspective camera model. *The Visual Computer*, 25(11):997–1008, 2009.

[108] Chao Jia and Brian L Evans. Online motion smoothing for video stabilization via constrained multiple-model estimation. *EURASIP Journal on Image and Video Processing*, 2017(1):1–13, 2017.

[109] Neel Joshi, Wolf Kienzle, Mike Toelle, Matt Uyttendaele, and Michael F Cohen. Real-time hyperlapse creation via optimal frame selection. *ACM Transactions on Graphics (TOG)*, 34 (4):1–9, 2015.

[110] Shuaicheng Liu, Ping Tan, Lu Yuan, Jian Sun, and Bing Zeng. Meshflow: Minimum latency online video stabilization. In *European Conference on Computer Vision*, pages 800–815. Springer, 2016.

[111] Brent Cardani. Optical image stabilization for digital cameras. *IEEE Control Systems Magazine*, 26(2):21–22, 2006.

[112] Fabrizio La Rosa, Maria Celvisia Virzì, Filippo Bonaccorso, and Marco Branciforte. Optical image stabilization (ois). *STMicroelectronics. Available online: http://www. st. com/resource/en/white_paper/ois_white_paper. pdf (accessed on 12 October 2017)*, 2015.

[113] Paresh Rawat and Jyoti Singhai. Review of motion estimation and video stabilization techniques for hand held mobile video. *Signal & Image Processing: An International Journal (SIPIJ) Vol*, 2, 2011.

[114] Sen-Zhe Xu, Jun Hu, Miao Wang, Tai-Jiang Mu, and Shi-Min Hu. Deep video stabilization using adversarial networks. *Computer Graphics Forum*, 37(7):267–276, 2018.

[115] Jinsoo Choi and In So Kweon. Deep iterative frame interpolation for full-frame video stabilization. *ACM Trans. Graph.*, 39(1):4:1–4:9, 2020.

[116] Azeddine Beghdadi, Mohamed-Chaker Larabi, Abdesselam Bouzerdoum, and Khan M. Iftekharuddin. A survey of perceptual image processing methods. *Sig. Proc.: Image Comm.*, 28(8):811–831, 2013.

[117] Derrington Andrew M., Allen Harriet A., and Louise S. Delicato. Visual mechanisms of motion analysis and motion perception. *Annu. Rev. Psychol*, 2007.

[118] Stevenson SB Yang Q Tiruveedhula P Roorda A. Arathorn, DW. How the unstable eye sees a stable and moving world. *Journal of Vision*, 13(10):1–19, 2013.

[119] Nick G Kingsbury. The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters. In *Proceedings of the IEEE Digital Signal Processing Workshop*, volume 86, pages 120–131, 1998.

[120] Shih-Hsuan Yang and Fu-Min Jheng. An adaptive image stabilization technique. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume 3, pages 1968–1973, 2006.

[121] Javier Sánchez and Jean-Michel Morel. Motion smoothing strategies for 2D video stabilization. *SIAM Journal on Imaging Sciences*, 11(1):219–251, 2018.

[122] Jiyang Yu and Ravi Ramamoorthi. Selfie video stabilization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 551–566, 2018.

Year	Authors	Motion estimation	Outlier detection	Motion model	Motion correction	Video synthesis	Advantages	Limitations
1996	Morimoto et al. [19]	block-matching	none	2D	static	dense	fast	very simple model
1997	Farid et al. [29]	pixel-based	none	2D	path-fitting	dense	fast	does not handle parallax
1998	Kingsbury et al. [119]	pixel-based	none	2D	filtering	dense	interpolated frames lower motion blur	no outlier rejection, does not handle parallax
2003	Litvin et al. [30]	pixel-based	none	2D	filtering	dense	reduces/eliminates resolution loss	does not handle parallax
2005	Matsushita et al. [32]	pixel-based	none	2D	filtering	dense	avoids resolution loss and blur	does not handle parallax, bad with moving objects
2006	Chang et al. [3]	pixel-based	frame-to-frame	2D	filtering	other	fast (real time)	does not handle parallax
	Tsoligkas et al. [33]	block-matching	frame-to-frame	2D	filtering	dense	fast	simple model
	Yang et al. [77]	features (SIFT)	video stream	2D	kalman filter	dense	handles abrupt motion	does not handle parallax
	Yang et al. [120]	block-matching	frame-to-frame	2D	path-fitting	dense	real-time	does not handle parallax
2007	Gleicher et al. [2]	features-based (SIFT)	none	2D	path-fitting	dense	fast, good video dynamics	low feature count/moving object cause problems
2008	Chen et al. [40]	features (SIFT)	frame-to-frame	2D	filtering	dense	reduces resolution loss	does not handle parallax
2009	Liu et al. [84]	features-based (KLT)	none	3D	path-fitting	sparse	handles parallax	SFM not robust
	Zhang et al. [107]	features (KLT)	video stream	3D	path-fitting	dense	full motion model	SFM slow and not robust
2011	Grundmann et al. [9]	pixel-based	frame-to-frame	2D	path-fitting	dense	stable, avoids sudden shifts in case of bad estimations	does not handle parallax, bad trade-off occur
	Liu et al. [21]	features-based (KLT)	frame-to-frame	perceptual	low-rank approximation	sparse	robust to motion parameters, handles parallax	does not handle moving objects
2012	Goldstein et al. [58]	features-based (KLT)	frame-to-frame	perceptual	filtering	CPW	2D/3D hybrid	needs good trajectories and large objects not handled
	Ringaby et al. [7]	features (KLT)	frame-to-frame	3D	filtering	dense	avoids SFM problems	model not always valid
2013	Liu et al. [71]	other	frame-to-frame	perceptual	path-fitting	sparse	robust and more precise than regular 2d	motion blur, large moving objects
	Wang et al. [37]	features (KLT)	video stream	perceptual	path-fitting	sparse	parallax without SFM	needs long trajectories, bad with foreground objects
	Rawat et al. [73]	pixel-based	none	2D	filtering	dense	handles abrupt motion	does not handle parallax
2014	Liu et al. [31]	pixel-based	video stream	perceptual	path-fitting	dense	handles parallax well	slow, dominant foreground
2015	Koh et al. [8]	features (KLT)	video stream	perceptual	path-fitting	CPW	handles dominant foreground	very specific case for outliers
2018	Wang et al. [99]	none	none	perceptual	CNN (learning)	dense	fast computation	necessity of annotated data for training
	Sanchez et al. [121]	pixel-based	frame-to-frame	2D	filtering	dense	good motion composition strategy	remains in 2D models
2019	Yu et al. [95]	pixel-based	none	2D	CNN (learning)	dense	no need for annotated data	computation time
2020	Choi et al. [115]	pixed-based	none	2D	CNN (learning)	dense	no need for annotated data	may introduce blur
2020	Zhao et al. [94]	pixed-based	none	perceptual	CNN (learning)	dense	flexible model that generalizes homographies	dependant on the training database

Table 1: Summary of the main approaches for video stabilization.

Year	Authors	Link	Number of videos	Categories	Ground truth
2009	Liu et al. [84]	web.cecs.pdx.edu/~fliu/project/3dstab.htm	32 (5)	N/A	no
2011	Grundmann et al. [9]	cc.gatech.edu/cpl/projects/videostabilization/	N/A	N/A	no
2011	Liu et al. [21]	web.cecs.pdx.edu/~fliu/project/subspace_stabilization/index.htm	109 (9)	N/A	no
2012	Goldstein et al. [58]	cse.huji.ac.il/~raananf/projects/stab/videos/	42 (42)	N/A	no
2013	Liu et al. [71]	liushuaicheng.org/SIGGRAPH2013/database.html	174 (174)	simple, quick rotation, zooming, large parallax, driving, crowd, running	no
2015	Koh et al. [8]	mcl.korea.ac.kr/stabilization/	162 (162)	simple, rolling shutter, depth, crowd, driving, running, object	no
2018	Yu et al. [122]	cseweb.ucsd.edu/~viscomp/projects/ECCV18VideoStab/	33 (33)	selfie	no
2018	Zhang et al. [96]	zhangleibit.github.io/download/stabfr/dataset.html	45 (45)	walking, climbing, running, riding, driving, large parallax, crowd, near-range object, dark	yes
2019	Ito et al. [98]	github.com/mariito/DVS_	421 (421)	simple, blurry, high motion, dark, textureless, parallax, discontinuous depth, crowd, close object	yes
2019	Wang et al. [99]	github.com/cxjyxxme/deep-online-video-stabilization-deploy	60 (60)	N/A	yes

Table 2: Summary of the main available databases for video stabilization. We display the number of videos reported in the original paper and the number of videos available online (in brackets)